FlexRAN Reference Solution NB-IOT L2-L1

API Specification

April 2018

Intel Confidential

Document Number: 575824-1.1

Contents

Figures

Tables

Revision History

Date	Revision	Description
April 2018	1.1	Correct typographical and format errors. Removed Intel® Transcede* brand-related description.
December 2017	1.0	Initial version based on the Intel® Transcede NBIOT API.

§

1 Introduction

This document describes the Intel FlexRAN reference for the physical layer (PHY) for a standalone Narrow Band—Internet of Things (NB-IoT) wireless base station using Intel® Xeon® processors. This API sits between the L2+ and the physical layer of a standalone NB-IoT base station.

This document describes the procedures supported by this API, command sequences involved on a per-procedure basis, command timing requirements, configuration parameters, and additional information required to know before integrating the L2+ stack with the PHY.

1.1 Acronyms

Table 1. Acronyms

Acronym	Definition
3GPP	3rd Generation Partnership Project
API	Application Program Interface
CNM	Continuous Network Monitor
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
CS	Cyclic Shift
CSI-RS	Channel State Indication Reference Signals
CTC	Convolutional Turbo Codes
DCI	Downlink Control Information
DFS	Dynamic Frequency Selection
DL	Downlink
DL-SCH	Downlink Shared Channel
DMA	Direct Memory Access

Acronym	Definition
DMRS	Demodulation Reference Symbol
E-UTRAN)	Evolved Universal Terrestrial Radio Access Network
eNB	eNodeB
HeMS	HeNB Management System
HeNB	Home Evolved Node B
LSB	Least Significant Bit
LTE	Long-Term Evolution
MAC	Media Access Control
MSB	Most Significant Bit
NB-IOT	Narrow Band—Internet of Things
NCCE	Narrowband Control Channel Element
NPDCCH	Narrowband Physical Downlink Control Channel
NPDSCH	Narrowband Physical Downlink Shared Channel
NPRACH	Narrowband Physical Random Access Channel
NPSS	Narrowband Primary Synchronization Signal
NPUSCH	Narrowband Physical Uplink Shared Channel
NRS	Narrowband Reference Signal
NSSS	Narrowband Secondary Synchronization Signal
PDCP	Packet Data Convergence Protocol
PHY	Physical Layer
RLC	Radio Link Control
RRC	Radio Resource Control
RNTI	Radio Network Temporary Identifier
SAP	Service Access Point
SON	Self-Organizing Network
TB	Transport Block
TBS	Transport Block Size

1.2 Reference Documents and Resources

A comprehensive library of Intel FlexRAN NB-IoT reference and application documentation is available from Intel. Developers, applications engineers, and other personnel who need to implement, support, or otherwise become familiar with the Intel FlexRAN NB-IoT reference PHY design can log on and register at My Intel to access the full range of support documentation. For detailed information on specific devices and available features, always consult the Intel® Business Link. Table 2 gives a summary of the Intel publications related to this document.

Table 2. Reference Documents and Resources

Title	Document Number/ Location
FlexRAN Reference Solution L1 XML Configuration User Guide	571741
FlexRAN Reference Solution Software v1.5.0 Release Notes	575822
FlexRAN Reference Solution NB-IOT User Guide	575823
MaxCore* Platform	https://www.artesyn.com/computing/products/product/max-core
Advanced Radio Tester Product Fact Sheet (PDF)	http://aceaxis.co.uk/website/wp-content/uploads/2016/07/AdvancedRadioTester_July2016.pdf
TM500 LTE TEST Mobile Application User Guide	https://www.aeroflex.com
Wind River® Linux Open Virtualization Profile, Virtual Node User's Guide, 6.0	www.windriver.com/
Wind River® Linux User Guide, 6.0	www.windriver.com/
Intel® C++ Compiler in Intel® Parallel Studio XE	https://software.intel.com/en-us/c-compilers/ipsxe
DPDK Documentation*	http://dpdk.org/doc/guides/

Table 3. 3GPP 36 Series Standard Document References

Ref.	Title	Number	Version
[36.211]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channel and Modulation	TS36.211	13.2.0
[36.212]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding	TS36.212	13.2.0
[36.213]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures	TS36.213	13.2.0
[36.214]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer; Measurements	TS36.214	13.2.0
[36.300]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Overall Description; Stage 2	TS36.300	13.2.0
[36.321]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control(MAC) Protocol Specification	TS36.321	13.2.0
[36.331]	3GPP Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control	TS36.331	13.2.0
[36.401]	3GPP Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Architecture Description	TS36.401	13.2.0

§

2 L1 API Overview

This document describes the FlexRAN NB-IoT eNB L2-L1 API, which provides the interface to the Long-Term Evolution (LTE) PHY running on an Intel® Xeon® processor device. The L2/L3 protocols expected to use this API are the radio resource control (RRC) layer [36.331], the media access control (MAC) layer [36.321], and the scheduler. This document assumes these also run on an Intel® Xeon® processor device on a different core. The NB-IoT L1 API described in this document currently supports NB-IoT standalone modes. The other two NB-IoT modes (in-band and guard band) are not described.

This document is divided into three main sections:

- **NB-IoT L1 API Procedures** provides a description of typical procedures that occur between the L1 and L2/L3 software.
- **NB-IoT L1 API Messages** provides the definition and encoding of the L1 API messages.
- **The API** facilitates a broad range of LTE functionality. Consult the relevant NB-IoT L1 release note to determine the actual functionality supported by the product.

2.1 L1 API

The L1 API, defined in this document, resides in the eNB component. The functionality of an eNB is shown in Figure 1 and Figure 2. In both figures, the location of the L1 API is highlighted.

Figure 1 shows the protocol model for the eNB defined in the E-UTRAN architectural standard [36.401]. It highlights the separation of control- and data-plane information, which is maintained throughout the NB-IoT network. Both control- and data-plane information are passed through the L1 API; however, each API message contains either control or data-plane information, but never both.

Figure 1. Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Protocol Model

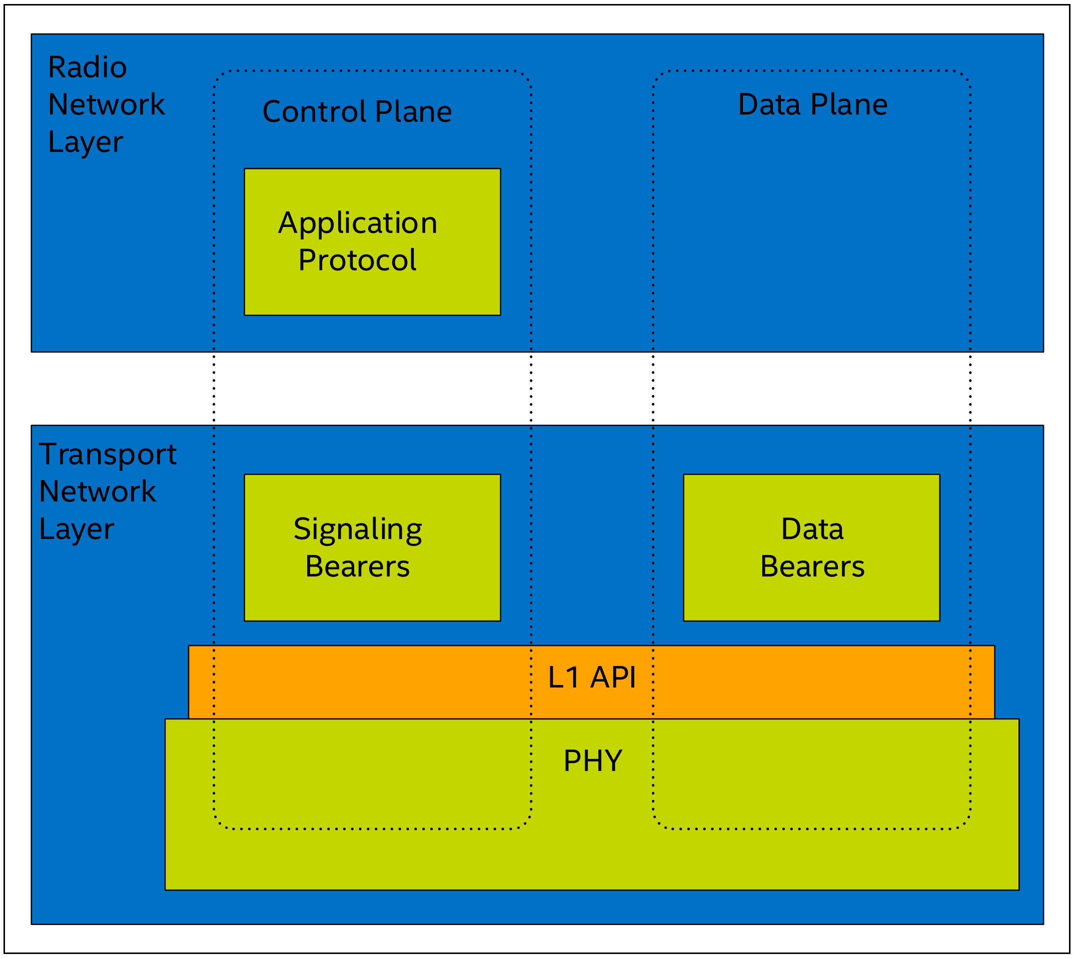
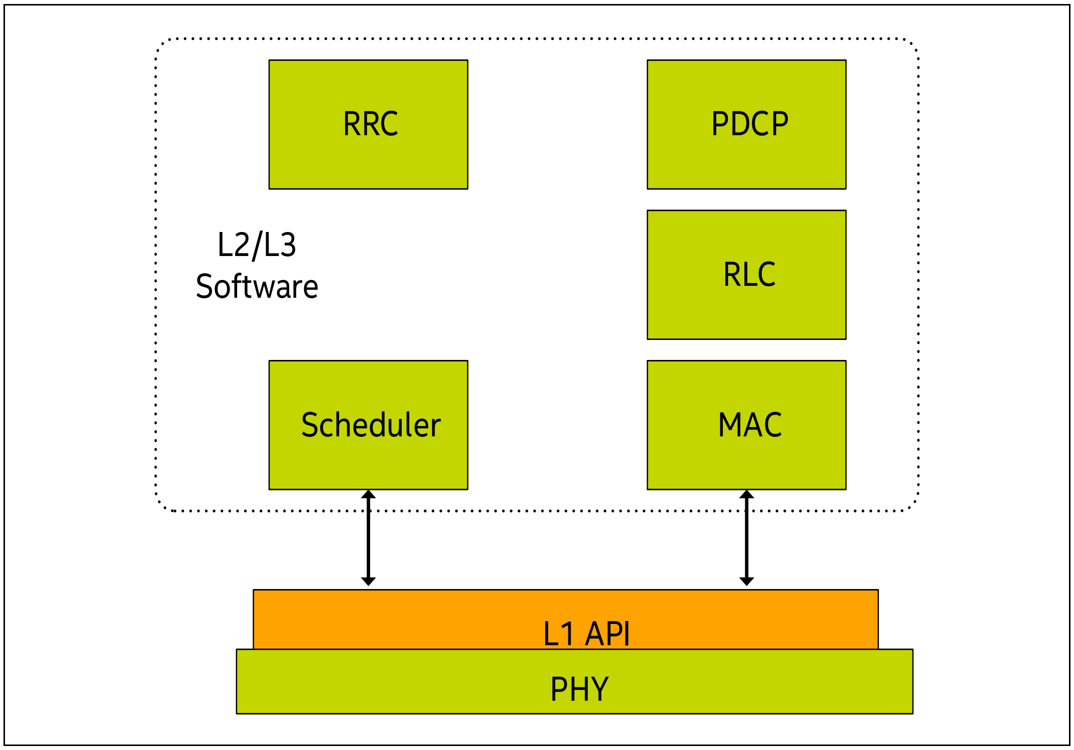


Figure 2 provides an example of how the different L2/L3 protocol layers interact with the L1 API. In this example, a PHY control entity is responsible for configuration procedures. The media access control (MAC) layer is responsible for the exchange of data-plane messages with the PHY.

The PHY configuration sent over the L1 interface may be determined using self-organizing network (SON) techniques, information model parameters sent from the HeNB management system (HeMS), or a combination of both methods. The MAC layer is responsible for the exchange of data-plane messages with the PHY. Finally, the scheduler is responsible for deciding the subframe structure and receiving measurement information from the PHY.

Note: There is no requirement for the eNB protocol to follow this example. The L1 API is a collection of messages that can be routed anywhere by the eNB software.

Figure 2. L1 API Interactions



§

3 L1 API Procedures

This section gives an overview of the procedures that use the L1 API. These procedures are split into two groups—configuration procedures and subframe procedures. Configuration procedures handle the management of the PHY layer and are expected to occur infrequently. Subframe procedures determine the structure of each 1ms subframe and operate with a 1ms periodicity.

3.1 Configuration Procedures

The configuration procedures supported by the L1 API are:

- Initialization
- Termination
- Restart
- Reset
- Reconfigure
- Query
- Notification

These procedures move the PHY layer through the IDLE, CONFIGURED, and RUNNING states, as shown in Figure 3. A list of the L1 API .request messages that are valid in each state is given in Table 4.

Figure 3. PHY Layer State Transactions on L1 API Configuration Messages

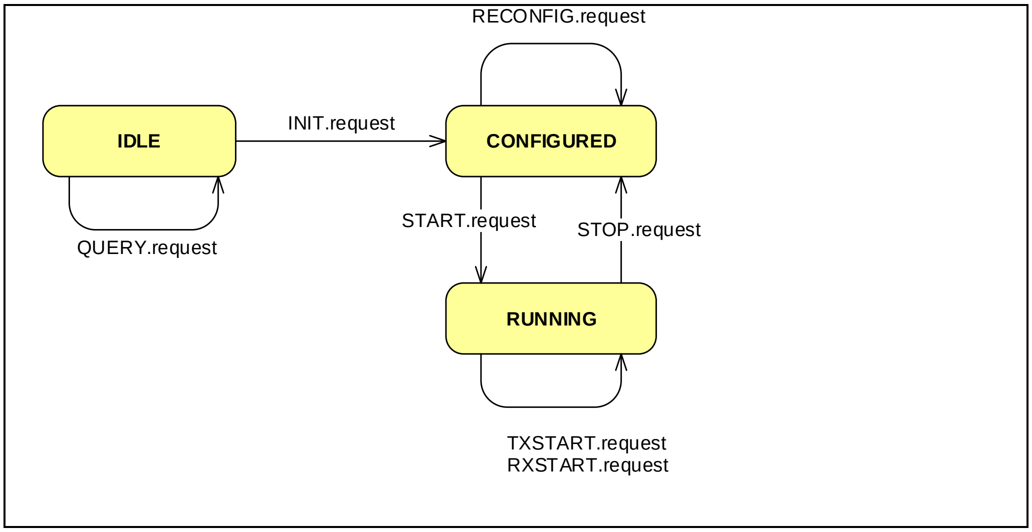


Table 4. L1 API Request Message Valid in Each PHY State

Idle State	Configured State	Running State
INIT.request	RECONFIG.request	STOP.request
	START.request	TXSTART.request
		RXSTART.request
QUERY.request		

3.2 Subframe Procedures

The subframe procedures have two purposes. First, they control the DL and UL frame structures. Second, they transfer the subframe data between the L2/L3 software and PHY. The subframe procedures supported by the L1 API are:

- Transmission of a 1ms SUBFRAME signal
- Synchronization of HyperSFN/SFN/SF between the L2/L3 software and PHY
- Transmission of the BCH transport channel carried by NPBCH
- Transmission of the PCH transport channel carried by NPDSCH

- Transmission of the DL-SCH transport channel carried by NPDSCH and reception of ACK/NACK response carried by Format2 NPUSCH
- Reception of the RACH transport channel carried by NPRACH
- Reception of the UL-SCH transport channel carried by Format1 NPUSCH and transmission of ACK/NACK response carried by NPDCCH
- RSSI measurements
- Received Interference Power Measurements

3.2.1 SUBFRAME Indication

A SUBFRAME indication is sent from the PHY, to the L2/L3 software, indicating the start of a 1ms frame. The actual subframe indication message used in Intel FlexRAN is the `PHY_TXSTART.indication` message.

3.2.2 API Message Order

The L1 API has constraints on when certain subframe messages can be sent, or will be received, by the L2/L3 software.

The downlink API message constraints are shown as below:

- The HyperSFN/SFN/SF issued by the PHY in the `SUBFRAME.indication` message is expected back in the corresponding `SUBFRAME.request` from the L2/L3.
- The `SUBFRAME.request` must be sent for every subframe and must be the next message. The `TXSTART.request` and `RXSTART.request` are mandatory, and the `TXSDU.request` is optional according to the channel type and channel state.
- There must be only one `TXSTART.request`, one `RXSTART.request`, and one `TXSDU.request` for a subframe.

The uplink API message constraints are as follows:

- If present, the NPUSCH format1 messages are delivered using `RXSDU.indications`.
- If present, the NPUSCH format2 message is delivered using `RXSTATUS.indication`.

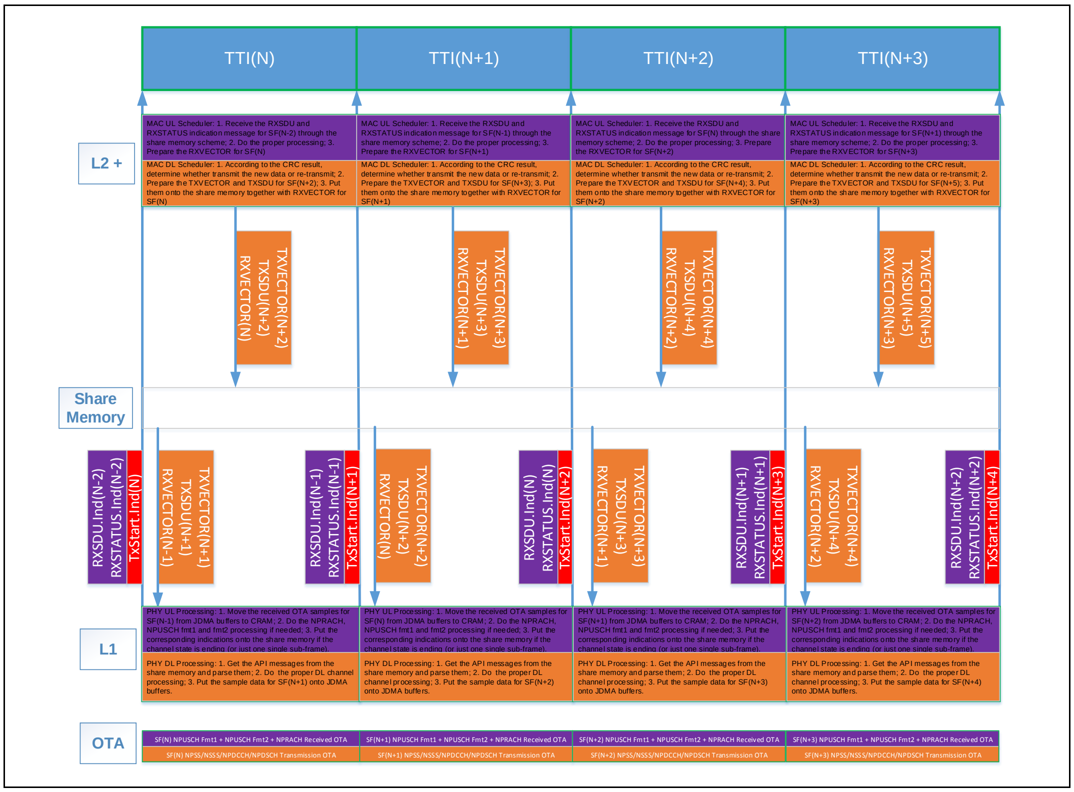
- If present, the NPRACH message is delivered using `RXSTATUS.indication`.
- There must be only one `RXSDU.indication` and one `RXSTATUS.indication` for a subframe.

There are as many ulPduReports in the RXSDU indication for NPUSCH format1 and ulHarqReports in the RXSTATUS indication for NPUSCH format2, as specified by the L2/L3 in the `RXSTART.request` messages. There could be multiple ulRachReports in the `RXSTATUS.indication` message, depending on the configurations of NPRACH and the signals received from the UEs.

3.2.3 API Message Timing

Figure 4 diagrams L1 API message timing.

Figure 4. API Message Timing



§

4 L1 API Messages

This section provides a description of the L1 API message formats. It defines the L1 API message header, message bodies, and error codes associated with the L1 API.

The L2/L3 and the PHY communicate by exchanging PHY SAP primitives. These primitives allow the L2/L3 to pass control information and data to be transmitted to the PHY and allow the PHY to pass status information and received data to the L2/L3. The control information passed with these primitives consists mainly of the subframe descriptor structures.

The protocol is message-based, with the maximum size of a message specified as part of the configuration of the interface.

The general message format of the L1 API is shown in Table 5, where it can be seen that each L1 API message consists of a header followed by a message body.

The generic header consists of 4 bytes, including a message type ID and message body length. Table 6 shows the current list of message types. The L1 API messages follow a standard naming convention where:

- All `.request` messages are sent from the L2/L3 software to the PHY.
- All `.confirmation` messages are sent from the PHY to the L2/L3 software. These are sent in response to a `.request`.
- All `.indication` messages are sent from the PHY to the L2/L3 software. These are sent asynchronously.

The message body is different for each message type; however, each message body obeys the following rules:

- The first field in each confirmation message is an error code. For each message, it indicates which error codes can be returned. A full list of error codes is given in Section 4.1.1 Message Type Field Coding.
- All messages should be aligned to 32-bit boundaries, and, if necessary, padding should be added to comply with this requirement.

A full description of each message body is given in the remainder of Section 4.3 L1 Configuration Messages.

4.1 Generic Message Header, Format, and Coding

All PHY service access point (SAP) primitives are translated into messages by prefixing the message header, as defined in Table 5. Each message header contains an appropriate message type field, uniquely identifying the PHY SAP primitive.

Table 5. LTE PHY SAP Message Header

Parameter	32-bit Word #	Bits	Length, Bits	Description
`phyEntityId`	0	31:24	8	PHY entity ID (Instance)
`msgType`	0	23:16	8	Message type
`msgSpecific`	0	15:0	16	Message payload length

The PHY entity (instance) ID is supplied by the control plane during the initialization to both L2 and PHY layers. This identifier allows the L2 to communicate with multiple PHY instances. Normally, the lower 16 bits of the header contain the length of the message in bytes. The message type values are known to both PHY and L2. The convention used for all primitives is based on a little-endian format for command fields. Only the actual payloads use a big-endian or network convention per 3GPP 36-series standards. This API only supports the little-endian format.

4.1.1 Message Type Field Coding

Table 6 specifies the message types associated with the primitives.

Table 6. Message Type Description

Value	Description	Section with Message Format
0	Reserved	N/A
1	PHY_TXSTART.request	4.4.1 PHY_TXSTART Request
2	Reserved	N/A
3	PHY_TXSTART.indication	4.4.3 PHY_TXSTART Indication
4	PHY_TXSDU.request	4.4.7.1 PHY_TXSDU Request
5	Reserved	N/A
6	Reserved	N/A

Value	Description	Section with Message Format
7	PHY_RXSTART.request	4.4.4 PHY_RXSTART Request
8	Reserved	N/A
9	Reserved	N/A
10	PHY_RXSDU.indication	4.4.8.1 PHY_RXSDU Indication
11	Reserved	N/A
12	PHY_INIT.request	4.3.1 PHY_INIT Request
13	PHY_INIT.indication	4.3.2 PHY_INIT Indication
14	PHY_RXSTATUS.indication	4.4.8.2 PHY_RXSTATUS Indication
15	PHY_RECONFIG.request	4.3.8 PHY_RECONFIG Request
16	PHY_RECONFIG.confirmation	4.3.9 PHY_RECONFIG Confirmation
17	PHY_START.request	4.3.3 PHY_START Request
18	PHY_START.confirmation	4.3.4 PHY_START Confirmation
19	PHY_STOP.request	4.3.5 PHY_STOP Request
20	PHY_STOP.confirmation	4.3.6 PHY_STOP Confirmation
21	PHY_STOP.indication	4.3.7 PHY_STOP Indication
22	Reserved	N/A
23	Reserved	N/A
24	Reserved	N/A
25	PHY_ERROR.indication	4.5.1 PHY_ERROR_IND
26	Reserved	N/A
27	Reserved	N/A
28	Reserved	N/A
29	Reserved	N/A
30	PHY_QUERY.request	4.3.10 PHY_QUERY Request
31	PHY_QUERY.indication	4.3.11 PHY_QUERY Indication
32–255	Reserved	N/A

4.2 Error Field Coding

In this document, the following error codes (ERRORCODE) are defined, to provide the cause of errors that occur in communications between the L2 and the PHY. The error code is generic and can be applied to applicable PHY SAP primitives. The ERRORCODE is an 8-bit value with the currently defined values listed in Table 7.

Table 7. Error Codes

Value	Description
0	Success
1	Primitive not supported (for requests)
2	FEC code type not supported
3	Overrun
4	Underrun
5	Transport Media Error
6	TX data size does not match TXVECTOR
7	Invalid TX VECTOR format
8	Invalid RX VECTOR format
9	RX CRC error
10	Transmit Timeout with missing TXSDU
11	Receiver did not match RXSDUs in RXVECTOR
12	Timeout on TXSTART.req
13	Timeout on RXSTART.req
14	No Custom Extensions Handler registered
15	L2/L3 rejection of CXR_STATE change command
16	L2/L3 rejection of CXR_FREQ_RECONFIQ request
16–255	Reserved

4.3 L1 Configuration Messages

4.3.1 PHY_INIT Request

Table 8. PHY_INIT Request

Parameter	32-bit Word #	Bits	Length, Bits	Description
phyEntityId	0	31:24	8	PHY entity ID (Instance)
msgType	0	23:16	8	Message type = 12
msgSpecific	0	15:0	16	PHY_INIT request payload length (INITPARM structure)
INITPARM structure	1 ~ 12	N/A	12X32	PHY_INIT parameters. Refer to Table 9.

This primitive is issued by L2 to request the PHY to perform an initialization process based on the parameters provided in the INITPARMS portion of this primitive.

Table 9. PHY INIT Parameters—Structure of INITPARM

Parameter	Length, Bits	Description, Value	Reference
operationModeInfo	4	Operation mode: 0 – Inband-SamePCI 1 – Inband-DifferentPCI 2 – guardband 3 – standalone 4 ~ 15 – Reserved	[36.331] Section 6.7.2 NOTE: Only supports value 3 – standalone mode.
nbPCI	12	Narrowband Physical Cell ID: 0 ~ 503	[36.211] Section 10.2.7
txAntNbr	8	NB-IoT transmitter antenna port count: {1, 2}	[36.211] Section 10.2.4.3
rxAntNbr	8	NB-IoT receiver antenna port count: {1}	[36.211] Section 10.1.3.5
subframePattern10	16	The set of valid subframes for downlink transmission within 10 ms:	[36.331] Section 6.7.2

Parameter	Length, Bits	Description, Value	Reference
		• The first/leftmost bit corresponds to the subframe #0 within one radio frame. • Value 0 in the bitmap indicates that the corresponding subframe is invalid for downlink transmission. • Value 1 in the bitmap indicates that the corresponding subframe is invalid for downlink transmission. • 10 least significant bits (LSBs) are used with the 6 most significant bits (MSBs) zero filled. • If enabled, subframePattern40 becomes reserved field bits.	
`subframePattern40`	3 x 16	The set of valid subframes for downlink transmission within 40 ms: • The first/leftmost bit corresponds to the subframe #0 within the radio frame satisfying SFN mod 4 = 0. • Value 0 in the bitmap indicates that the corresponding subframe is invalid for downlink transmission. • Value 1 in the bitmap indicates that the corresponding subframe is invalid for downlink transmission. • 40 LSB bits (subframePattern40[0], subframePattern40[1] and 8 LSB bits of subframePattern40[2]) are actually used with the 8 MSB's zero filled in the last word. • If enabled, subframePattern10 becomes reserved field bits.	[36.331] Section 6.7.2
`nrsPwr`	16	Narrowband reference signal (NRS) power control: • It uses two's complement notation in Q8 dB steps, so the range is from -128 to 6 dBr, where 0 dBr is equal to a two's complement	[36.213] Section 16.2.2 [36.331] Section 6.7.2

Parameter	Length, Bits	Description, Value	Reference
		rms level of 1 in the IQ samples of each antenna. **Notes:** (a) The IQ samples are stored as signed 16-bits I and 16-bits Q. So, the fixed-point format has 12 fractional bits (1:3:12). (b) In [36.331] Section 6.7.2, nrsPower provides the downlink narrowband reference-signal EPRE, which is the true value in dBm. The range is from -60 to 50 dBm. So when we use this parameter in our physical layer software and the higher layer software, we need to figure out the relationship between the dBr and real power dBm value according to the RF gain. (c) The ratio of NPDSCH EPRE to NRS EPRE is 0 dB when txAntNbr = 1, -3 dB when txAntNbr = 2. (d) The ratio of NPBCH EPRE to NRS EPRE is 0 dB when txAntNbr = 1, -3 dB when txAntNbr = 2. (e) The ratio of NPDCCH EPRE to NRS EPRE is 0 dB when txAntNbr = 1, -3 dB when txAntNbr = 2;	
`nprachFormat`	1	NPRACH preamble format: 0 – Cyclic prefix length for NPRACH transmission is 66.7us. 1 – Cyclic prefix length for NPRACH transmission is 266.7us (this is the only value currently supported).	[36.211] Section 10.1.6.1 [36.331] Section 6.7.2
`nprachConfigurationNum`	2	Number of NPRACH configurations: {1, 2, 3}	N/A
Reserved1	13	Reserved	N/A
`nprachConfigurationArray` (three structures of `tNprachConfiguration`)	3X1X32	Refer to Table 10 for more details.	N/A

Parameter	Length, Bits	Description, Value	Reference
ndpcchMaxNumRepetitionPagingIndex	4	Maximum number of repetitions for NPDCCH type1 common search space (CSS) for paging: 0 – 1; 1 – 2; 2 – 4; 3 – 8; 4 – 16; 5 – 32; 6 – 64; 7 – 128; 8 – 256; 9 – 512; 10 – 1024; 11 – 2048	[36.213] Section 16.6 [36.331] Section 6.7.2
Reserved2	12	Reserved	N/A
dlGapFlag	1	Flag to indicate whether the higher layer configures the downlink gap configuration.	[36.331] Section 6.7.2
dlGapThresholdIndex	2	Threshold on the maximum number of repetitions configured for NPDCCH before application of DL transmission gap configuration: 0 – 32; 1 – 64; 2 – 128; 3 – 256	[36.331] Section 6.7.2
dlGapPeriodicityIndex	2	Periodicity of a DL transmission gap in number of subframes: 0 – 64 subframes; 1 – 128 subframes. 2 – 256 subframes; 3 – 512 subframes.	[36.331] Section 6.7.2
dlGapDurationCoeff	2	Coefficient to calculate the gap duration of a DL transmission: 0 – 1/8; 1 – 1/4; 2 – 3/8; 3 – 1/2	[36.331] Section 6.7.2
Reserved3	9	Reserved	N/A
srsSubframeConfig	4	SRS subframe configuration: {0, 1, ..., 15}	[36.211] Section 5.5.3.3 Table 5.5.3.3-1 Table 5.5.3.3-2
groupHoppingEnabled	1	Flag to indicate to enable the group hopping: {0, 1}	[36.211] Section 10.1.4.1.3
Reserved4	11	Reserved	N/A
eutraNumCRSPorts	1	Number of E-UTRA CRS antenna ports: 0 – The same number of ports as NRS 1 – 4 antenna ports	[36.331] Section 6.7.2

Parameter	Length, Bits	Description, Value	Reference
`eutraCtrlRegionSize`	2	LTE control region size: {1, 2, 3}	[36.331] Section 6.7.2
`nrsCRSPwrOffset`	5	Power offset from NRS to LTE CRS: 0 – -6dB; 1 – -4.77dB; 2 – -3dB; 3 – -1.77dB; 4 – 0dB; 5 – 1dB; 6 – 1.23dB; 7 – 2 dB; 8 – 3dB; 9 – 4dB; 10 – 4.23dB; 11 – 5dB; 12 – 6dB; 13 – 7dB; 14 – 8dB; 15 – 9dB; 31 – This parameter is not provided by higher layer, so the ratio of NRS. EPRE to CRS EPRE will be assumed to be 0dB. 16 ~ 30 is reserved. **Note:** This parameter is only valid when the operationModeInfo indicates '00'.	[36.331] Section 6.7.2
`rasterOffset`	3	NB-IoT offset from LTE channel raster: 0 – -7.5 kHz 1 – -2.5 kHz 2 – +2.5 kHz 3 – +7.5 kHz	[36.331] Section 6.7.2
`eutraCRSSequenceInfo`	5	Index information that indicates the index of the PRB containing PSS/NSSS/NPBCH: {0, 1, ..., 31}	[36.331] Section 6.7.2
`crsPwr`	16	LTE cell-specific reference signal (CRS) power control uses two's complement notation in Q8 dB steps, so the range is from -128 to 6 dBr, where 0 dBr is equal to a two's complement rms level of 1 in the IQ samples of each antenna. **Notes:** (a) The IQ samples are stored as signed 16-bits I and 16-bits Q. So, the fixed-point format has 12 fractional bits (1:3:12)	N/A

Parameter	Length, Bits	Description, Value	Reference
		(b) In [36.331] Section 6, crsPower provides the downlink narrowband reference-signal EPRE, which is the true value in dBm. And the range is from -60 to 50 dBm. So when we use this parameter in our physical layer software and the higher layer software, we need to figure out the relationship between the dBr and real power dBm value according to the RF gain. (c) This parameter is only valid when the operationModeInfo indicates '00'.	
`hyperFrameNumber`	16	NB-IoT initialized hyper radio frame number: {0…1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.	[36.331] Section 6.7.2
`frameNumber`	16	NB-IoT initialized radio frame number: {0…1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.	[36.331] Section 6.7.2
`subFrameNumber`	8	NB-IoT initialized subframe number: {0…9}, 4 LSB bits are actually used with the 4 MSB's zero filled.	[36.331] Section 6.7.2
`scaleFactorOfSampleRate`	8	The scale factor of the sample rate relative with Ts under 30.72MS/s: {1, 2, 4, 8, 16} The default value is 16.	N/A
`phyCfg`	32	• The default value is 263. • The phyCfg allows backward compatibility with previous releases. • Currently, LSB is used. • Bit 0 Set: Use FIFO-based MAC-PHY interface. • Bit 1 Set: Do automatic processing of PBCH within PHY ignoring MAC BCH messages. • Bit 2 Set: PHY robust processing enabled. When enabled, the PHY, in the event that the MAC layer does not issue the `TXVECTOR` and	N/A

Parameter	Length, Bits	Description, Value	Reference
		`RXVECTOR` on time, injects dummy TX and RX Vectors with zero channels and also when needed transmits the PBCH information. In the case of missing TX and RX vectors, it issues to the MAC error codes `TIMEOUT_RXSTART_REQ` and `TIMEOUT_TXSTART_REQ`. Also, on the receiver RXSDU's and RXSTATUS with this error are issued for every expected one in the late `rxvector` allowing the MAC to free any buffers that have been previously allocated. • Bit 3 Set: PHY uses TXSDU pointer to data message instead of data lumped together with the `txsdu` request. In this case, the PHY frees both the memory associated with the `TxSduReq` message and the memory associated with the payload of the message after the processing has completed. Using this scheme allows the MAC to save a memory copy. This setting can be overruled by the setting of bit 9. If bit 9 is zero, the previously described behavior is true, else bit 9 defines the contents of the pointer. • Bit 4 Set: PHY uses a prioritized Harq ACK/NACK list for control data that is multiplexed on the PUSCH channel over the HARQ/RI information to minimize HARQ latency. The first one contains only the HARQ ACK/NACK information, the second list is for the PUSCH Received data, and there is even a third lower priority list for CQI and RI information. • Bit 5 Set: PHY will not free next missed vector. This option is useful when the L2/L3 implements its own logic to detect a late vector and then it doesn't push the late messages to the FIFO. If the bit is clear, the PHY, after detecting a late message being	

Parameter	Length, Bits	Description, Value	Reference
		delivered from the L2/L3, will pop and free all those messages as invalid. • Bit 6 Set: Use Intra TTI Indications. Intra TTI indications are issued between subframes to allow the L2/L3 to implement robust handling of late APIs from the PHY. • Bit 7 Set: No SDUs are sent if SR or RACH are not detected. • Bit 8 Set: Configures PHY for low latency path which, in conjunction with matching configuration in L2/L3 stack, allows for 8 ms DL HARQ Latency • Bit 9 Set: `TxSDU` pointer is a pointer to an array of pointers and lengths for channel Type PDSCH only. When clear, bit 3 defines the meaning of the `TxSDU` pointer. • Bit 10: Reserved. • Bit 11: Reserved. • Bit 12: IRC global enable. When this bit is set, a PUSCH channel can be configured individually to use the IRC instead of the MRC default scheme using the `ULChannelDescriptor ircEnable` bit. When this bit is clear, the IRC algorithm is always disabled, regardless of the state of the `ULChannelDescriptor ircEnable` bit. • Bits 13–31: Reserved.	

Table 10. NPRACH Configuration Parameters—Structure of tNprachConfiguration

Parameter	Length, Bits	Description, value	Reference
`nprachPeriodicityIndex`	3	NPRACH resource periodicity in the unit ms: 0 – 40; 1 – 80; 2 – 160; 3 – 240; 4 – 320; 5 – 640; 6 – 1280; 7 – 2560	[36.211] Section 10.1.6.1 [36.331] Section 6.7.2

Parameter	Length, Bits	Description, value	Reference
nprachStartTimeIndex	3	NPRACH starting time in the unit ms: 0 – 8; 1 – 16; 2 – 32; 3 – 64; 4 – 128; 5 – 256; 6 – 512; 7 – 1024	[36.211] Section 10.1.6.1 [36.331] Section 6.7.2
nprachSubcarrierOffsetIndex	3	Frequency location of the first subcarrier allocated to NPRACH: 0 – 0; 1 – 12; 2 – 24; 3 – 36; 4 – 2; 5 – 18; 6 – 34	[36.211] Section 10.1.6.1 [36.331] Section 6.7.2
nprachNumSubcarriersIndex	2	Number of subcarriers allocated to NPRACH: 0 – 12; 1 – 24; 2 – 36; 3 – 48	[36.211] Section 10.1.6.1 [36.331] Section 6.7.2
nprachSubcarrierRangeStartIndex	2	Fraction for calculating starting subcarrier index for the range of NPRACH subcarriers reserved for indication of UE support for multitone msg3 transmission: 0 – 0; 1 – 1/3; 2 – 2/3; 3 – 1	[36.211] Section 10.1.6.1 [36.331] Section 6.7.2
nprachNumRepetitionsIndex	3	Number of NPRACH repetitions per attempt: 0 – 1; 1 – 2; 2 – 4; 3 – 8; 4 – 16; 5 – 32; 6 – 64; 7 – 128	[36.211] Section 10.1.6.1 [36.331] Section 6.7.2
npdcchMaxNumRepetitionsRAIndex	4	Maximum number of repetitions for NPDCCH type2 common search space (CSS) for RAR, Msg3 retransmission and Msg4: 0 – 1; 1 – 2; 2 – 4; 3 – 8; 4 – 16; 5 – 32; 6 – 64; 7 – 128; 8 – 256; 9 – 512; 10 – 1024; 11 – 2048	[36.213] Section 16.6 [36.331] Section 6.7.2
npdcchStartSFType2CSSIndex	3	Starting subframe configuration for NPDCCH type2 common search space (CSS) for RAR, Msg3 retransmission and Msg4: 0 – 3/2; 1 – 2; 2 – 4; 3 – 8; 4 – 16; 5 – 32; 6 – 48; 7 – 64	[36.213] Section 16.6 [36.331] Section 6.7.2

Parameter	Length, Bits	Description, value	Reference
npdcchStartSFOffsetType2CSSIndex	2	Fractional period offset of starting subframe for NPDCCH type2 common search space (CSS) for RAR, Msg3 retransmission and Msg4: 0 – 0; 1 – 1/8; 2 – 1/4; 3- 3/8	[36.213] Section 16.6 [36.331] Section 6.7.2
Reserved	7	Reserved	N/A

4.3.2 PHY_INIT Indication

This primitive is issued by the PHY to inform the L2 that it has completed the initialization process requested by the `PHY_INIT.request`. Table 11 shows the PHY_INIT indication details.

Table 11. PHY_INIT Indication

Parameter	32-bit Word #	Bits	Length, Bits	Description
phyEntityId	0	31:24	8	PHY entity ID (Instance)
msgType	0	23:16	8	Message type = 13
status	0	15:8	8	Status 0 = Success Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)
Reserved	0	7:0	8	Reserved

4.3.3 PHY_START Request

This primitive is issued by the L2 to request the PHY to start or resume processing depending on whether a `PHY_INIT` or a `PHY_RECONFIG` command was issued earlier.

The purpose of this command is for the PHY to synchronize with the Radio Interface Frame clock and start issuing subframe indications to the L2 so the normal subframe command exchange described in Section 3.2.2 API Message Order can take place. Table 12 shows the `PHY_START` request details.

Table 12. PHY_START Request

Parameter	32-bit Word #	Bits	Size, Bits	Description
`phyEntityId`	0	31:24	8	PHY entity ID (Instance)
`msgType`	0	23:16	8	Message Type = 17
`msgSpecific`	0	15:0	16	Message Length
`phyEntityId`	1	31:24	8	PHY entity ID (Instance)
`msgType`	1	23:16	8	Message Type = 17
`mode`	1	15-0	16	Operational Mode (based on interrupt source and radiohead interface type): 0 – CPRI 1 – Timer-based (used for debugging only) 3 – External event (used for debugging only) 4 – PCIe (used for RF module interface) (default)
`count`	2	31:0	32	CPRI: 0 Normal Mode, 2 RRH mode Timer: Stop after count subframes External: N/A
`period`	3	31:0	32	Resolution (related to operational mode) CPRI: N/A Timer: Period in ms (default is 1) External: N/A

4.3.4 PHY_START Confirmation

This primitive is issued by the PHY to acknowledge to L2 that it has started the synchronization with the radio interface as a result of the receipt of the `PHY_START.request` command. The `TXSTART.indication` issued to the L2 confirms that the synchronization has been achieved. Table 13 shows the `PHY_START` confirmation details.

Table 13. PHY_START Confirmation

32-bit Word #	Bits	Size, Bits	Description
0	31:24	8	PHY entity ID (Instance)
0	23:16	8	Message type = 18
0	15:8	8	Status 0 = Success Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)
0	7:0	8	Subframe number

4.3.5 PHY_STOP Request

This primitive is issued by the L2 to request the PHY to stop subframe processing. The purpose of this command is for the PHY to disable the processing of the radio interface timing-derived interrupts to allow its reconfiguration with a `PHY_RECONFIG.request` command. Table 14 shows the `PHY_STOP` request details.

Table 14. PHY_STOP Request

32-bit Word #	Bits	Size, Bits	Description
0	31:24	8	PHY entity ID (Instance)
0	23:16	8	Message type = 19
0	15-0	16	Message length

4.3.6 PHY_STOP Confirmation

This primitive is issued by the PHY to inform the L2 that it completed the execution of the `PHY_STOP.request` command. Depending on the internal PHY state, the `PHY_STOP` confirmation can be delayed until completion of the processing of the ongoing subframe. Table 15 shows the `PHY_STOP` confirmation details.

Table 15. PHY_STOP Confirmation

32-bit Word #	Bits	Size, bits	Description
0	31:24	8	PHY entity ID (Instance)
0	23:16	8	Message type = 20
0	15:8	8	Status 0 = Success Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)
0	7:0	8	Reserved

4.3.7 PHY_STOP Indication

The `PHY_STOP` Indication is sent when the PHY automatically stops processing according to `PHY_START.request` parameters. For example, if `PHY_START.request` specifies `Count = 100` subframes and `Mode = CPRI`, then PHY automatically stops after 100 loops and this indication is sent. Table 16 shows the `PHY_STOP` Indication details.

Table 16. PHY_STOP Indication

32-bit Word #	Bits	Size, bits	Description
0	31:24	8	PHY entity ID (Instance)
0	23:16	8	Message type = 21
0	15:8	8	Status 0 = Success Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)
0	7:0	8	Subframe number

4.3.8 PHY_RECONFIG Request

This primitive is issued by the L2 to request the PHY to perform a reconfiguration process based on the parameters provided in the `INITPARMS` portion of this primitive. This command is to be issued by the L2 once it has issued a `PHY_STOP` and received the `PHY_STOP` confirmation from the PHY. Table 17 shows the `PHY_RECONFIG` request details.

Table 17. PHY_RECONFIG Request

32-bit Word #	Bits	Size, Bits	Description
0	31:24	8	PHY entity ID (Instance)
0	23:16	8	Message type = 15
0	15-0	16	`PHY_INIT` payload length (`INITPARM` structure)
…	…	…	Parameters

4.3.9 PHY_RECONFIG Confirmation

This primitive is issued by the PHY to inform the L2 that it has completed the reconfiguration process requested by the `PHY_RECONFIG.request` command. Table 18 shows the `PHY_RECONFNIG` confirmation details.

Table 18. PHY_RECONFIG Confirmation

32-bit Word #	Bits	Size, Bits	Description
0	31:24	8	PHY entity ID (Instance)
0	23:16	8	Message type = 16
0	15:8	8	Status 0 = Success Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)
0	7:0	8	Reserved

4.3.10 PHY_QUERY Request

The `PHY_QUERY.request` is used by the L2/L3 software to determine the API version supported by the PHY. The version for any PHY release that incorporates new features, enhancements, or changes that affect the API can be found in the release notes provided with the PHY release. Table 19 shows the `PHY_QUERY` request details.

Table 19. PHY_QUERY Request

32-bit Word #	Bits	Size, Bits	Description
0	31:24	8	PHY entity ID (Instance)
0	23:16	8	Message type = 30
0	15-0	16	Message length

4.3.11 PHY_QUERY Indication

The `PHY_QUERY.indication` primitive is issued by the PHY to the L2/L3 software with a text string that represents the version of the current PHY code. If the version information is not a multiple of 4 bytes, bytes containing zero are appended to preserve 32-bit alignment. Table 20 shows the `PHY_QUERY` Indication details.

Table 20. PHY_QUERY Indication

32-bit Word #	Bits	Size, Bits	Description
0	31:24	8	PHY entity ID (Instance)
0	23:16	8	Message type = 31
0	15-0	16	Message length
1	31:0	32	Version String Bytes 3..0
…		32	…
n	31:0	32	Version String Bytes n/4, n-1/4, n-2/4, n-3/4

4.4 L1 Subframe Messages

4.4.1 PHY_TXSTART Request

This primitive, detailed in Table 21, is issued by the L2 to request the eNB PHY to start the transmission of PHY PDUs in the nearest DL subframe, and carries all the control information needed for the local PHY as described in Table 22.

As a result of the reception of this primitive, the PHY sends a confirmation primitive, and at the proper moment, starts transmission of the PHY PDUs or performs requested actions as defined by the control information contained in `TXVECTOR`. The `TXVECTOR` corresponds to the DL-Scheduling Information currently being sent to remote UE. The `TXVECTOR` contains only the DL-Scheduling Information parameters that are necessary for the PHY to perform the correct transmit actions.

Table 23 shows the NPDCCH configuration parameters for the Structure of `tNpdcchConfiguration`, and Table 24 provides the NPDSCH Configuration parameters for the structure of `tNpdschConfiguration`.

Table 21. PHY_TXSTART Request

Parameter	32-bit Word #	Bits	Size, bits	Description
`phyEntityId`	0	31:24	8	PHY entity ID (Instance)
`msgType`	0	23:16	8	Message type = 1
`msgSpecific`	0	15:0	16	`PHY_TXSTART` request payload length (`DLSUBFRDESC` structure)
`DLSUBFRDESC` structure	N/A	N/A	N/A	`PHY_TXSTART` parameters. (Refer to Table 22.)

Table 22. DL Subframe Descriptor Parameters—Structure of DLSUBFRDESC

Parameter	Length, Bits	Description, Value	Reference
`hyperFrameNumber`	16	NB-IoT downlink hyper radio frame number: {0...1023}, 10 LSB bits are actually used with the 6 MSBs zero filled	[36.331] Section 6.7.2

Parameter	Length, Bits	Description, Value	Reference
`frameNumber`	16	NB-IoT downlink radio frame number: {0…1023}, 10 LSB bits are actually used with the 6 MSBs zero filled	[36.331] Section 6.2.2
`subFrameNumber`	8	NB-IoT downlink subframe number: {0…9}, 4 LSB bits are actually used with the 4 MSBs zero filled	[36.331] Section 6.2.2
`chanType`	4	NB-IoT downlink channel type: 0 – NPSS, it is only valid when subFrameNumber is 5 1 – NSSS, it is only valid when subFrameNumber is 9 and frameNumber mod 2 = 0 2 – NPBCH, it is only valid when subFrameNumber is 0 and frameNumber mod 64 = 0 3 – NPDCCH 4 – NPDSCH 8 – Empty 5 ~ 7 – Reserved 9 ~ 15 – Reserved	N/A
`chanState`	4	State of nbDlChannel: 0 – Just begin at this subframe 1 – To be continuous at this subframe 3 – To be end at this subframe 2, 4 ~ 15 – Reserved	N/A
`npdcchNumber`	2	Number of NPDCCH transmitted within this subframe: {0, 1, 2}	N/A
`npdschNumber`	2	Number of NPDCCH transmitted within this subframe: {0, 1}	N/A
`lStartSymbol`	2	For NPSS, lStartSymbol = 3; For NSSS, lStartSymbol = 3; For NPBCH, lStartSymbol = 3;	[36.211] Section 10.2.4

Parameter	Length, Bits	Description, Value	Reference
		For NPDCCH, lStartSymbol = eutraControlRegionSize when operationModeInfo is 0 or 1, lStartSymbol = 0 when operationModeInfo is 2 or 3 For NPDSCH carrying SIB1-NB, lStartSymbol = 3 when operationModeInfo is 0 or 1, lStartSymbol = 0 when operationModeInfo is 2 or 3; For other NPDSCH, lStartSymbol = eutraControlRegionSize when operationModeInfo is 0 or 1, lStartSymbol = 0 when operationModeInfo is 2 or 3	[36.211] Section 10.2.7 [36.213] Section 16.4.1.4 [36.213] Section 16.6.1
`modScheme`	4	Modulation scheme: 1 – QPSK, which is only valid for NPBCH, NPDCCH or NPDSCH 0, 2 ~ 15 – Reserved	[36.211] Section 10.2.3-5
`transimissionScheme`	4	Transmission scheme: 0 – Single-antenna port scheme 1 – Transmit diversity scheme 2 ~ 15 is reserved	[36.213] Section 16.4.1
`offsetNpdschConfiguration`	32	Offset from the beginning of the structure to the element npdschConfigurationArray. Since this entry is optional, a value of zero means that this entry is absent for this subframe.	N/A
`npdcchConfigurationArray` (`npdcchNumber` structures of `tNpdcchConfiguration`)	npdcchNumber X1X32	Refer to Table 23 for more details.	N/A
`npdschConfigurationArray`	npdcchNumber X2X32	Refer to Table 24 for more details	N/A

Parameter	Length, Bits	Description, Value	Reference
(npdschNumber structures of tNpdschConfiguration)			

Table 23. NPDCCH Configuration Parameters—Structure of tNpdcchConfiguration

Parameter	Length, Bits	Description, Value	Reference
npdcchNcceIndex	2	NCCE Index that this NPDCCH uses: {0, 1}	[36.211] Section 10.2.5.1
npdcchSearchSpaceType	2	NPDCCH search space type: 0 – Type1-NPDCCH common search space 1 – Type2-NPDCCH common search space 2 – UE-specific search space 3 – Reserved	[36.213] Section 16.4.1
npdcchFormatType	2	NPDCCH format type: 0 – NPDCCH format0 that will use 1 NCCE 1 – NPDCCH format1 that will use 2 NCCEs 2 ~ 3 – Reserved	[36.211] Section 10.2.5.1
npdcchRepetitionLevelIndex	3	NPDCCH repetition level index: {0, 1, ..., 7} This parameter determines NPDCCH repetition level according to [36.213] Table 16.6-1, Table 16.6-2, and Table 16.6.3 together with the search space type and Rmax.	[36.213] Section 16.6 Table 16.6-1 Table 16.6-2 Table 16.6-3
npdcchMaxNumRepetitionsIndex	4	Maximum number of repetitions for NPDCCH: 0 – 1; 1 – 2; 2 – 4; 3 – 8; 4 – 16; 5 – 32; 6 – 64; 7 – 128; 8 – 256; 9 – 512; 10 – 1024; 11 – 2048	[36.213] Section 16.6 [36.331] Section 6.7.2
npdcchStartSFIndex	3	Starting subframe configuration for NPDCCH: 0 – 3/2; 1 – 2; 2 – 4; 3 – 8; 4 – 16; 5 – 32; 6 – 48; 7 – 64	[36.213] Section 16.6 [36.331] Section 6.7.2

Parameter	Length, Bits	Description, Value	Reference
`npdcchStartSFOffsetIndex`	2	Fractional period offset of starting subframe for NPDCCH: 0 – 0; 1 – 1/8; 2 – 1/4; 3- 3/8	[36.213] Section 16.6 [36.331] Section 6.7.2
Reserved1	14	Reserved	N/A
`nRNTI`	16	Radio network temporary identifier: {0, 1…65535}	[36.321] Section 7.1 Table 7.1-2
Reserved2	16	Reserved	N/A

Table 24. NPDSCH Configuration Parameters—Structure of tNpdschConfiguration

Parameter	Length, Bits	Description, Value	Reference
`npdschType`	4	NB-IoT NPDSCH type: 4 – NPDSCH carrying others 5 – NPDSCH carrying SIB1-NB 6 – NPDSCH carrying SIBx-NB except for SIB1-NB 7 – NPDSCH carrying paging 0 ~ 3 – Reserved 8 ~ 15 – Reserved	N/A
`schedulingInfoSib1OrImcs`	4	Scheduling information for NPDSCH carrying SIB1-NB or modulation and coding scheme for other NPDSCH: {0, 1, …, 15} This parameter is only valid for NPDSCH. As for NPDSCH carrying SIB1-NB, this parameter not only determines the transport block size (TBS) for SIB1-NB, but it also determines the number of NB-SIB1 repetitions and the starting radio frame number for NB-SIB1 repetitions. As for NPDSCH carrying non-SIBx-NB, this parameter determines the transport block size (TBS) together with `resourceAssignmentIndex`.	[36.213] Section 16.4.1.5 Table 16.4.1.5.1-1 Table 16.4.1.5.2-1 [36.213] Section 16.4.1.3 Table 16.4.1.3-3 Table 16.4.1.3-4

Parameter	Length, Bits	Description, Value	Reference
npdcchMaxNumRepetitionsIndex	4	Maximum number of repetitions for NPDCCH used to schedule this NPDSCH: 0 – 1; 1 – 2; 2 – 4; 3 – 8; 4 – 16; 5 – 32; 6 – 64; 7 – 128; 8 – 256; 9 – 512; 10 – 1024; 11 – 2048	[36.213] Section 16.6 [36.331] Section 6.7.2
repetitionNumberIndex	4	Repetition number index: {0, 1, ..., 15} This parameter is only valid for NPDSCH carrying non-SIBx-NB, and it can determine a repetition number for NPDSCH carrying non-SIBx-NB, according to [36.213] Section 16.4.1.3 Table 16.4.1.3-2.	[36.213] Section 16.4.1.3 Table 16.4.1.3-2
resourceAssignmentIndex	3	Resource assignment index: {0, 1, ..., 7} This parameter is only valid for NPDSCH carrying non-SIBx-NB. It can determine a number of subframes that NPDSCH is mapped to according to [36.213] Section 16.4.1.3 Table 16.4.1.3-1.	[36.213] Section 16.4.1.3 Table 16.4.1.3-1
newDataIndicator	1	New data indicator: {0, 1}	[36.212] Section 6.4.3
schedulingDelayIndex	3	Scheduling delay index: {0, 1, ..., 7} This parameter is only valid for NPDCCH with DCI format N1, which determines the first scheduling delay value k0 according to [36.213]. Table 16.4.1-1. For NPDCCH with DCI format N2, the k0 = 5.	[36.213] Section 16.4.1 Table 16.4.1-1
siPeriodicityIndex	3	Periodicity index of the SI-message in radio frames: 0 – rf64; 1 – rf128; 2 – rf256; 3 – rf512; 4 – rf1024; 5 – rf2048; 6 – rf4096	[36.331] Section 6.7.2
siRepetitionPatternIndex	2	Indicates the starting radio frames within the SI window used for SI message transmission: 0 – every2ndRF;1 – every4thRF; 2 – every8thRF; 3 – every16thRF	[36.331] Section 6.7.2

Parameter	Length, Bits	Description, Value	Reference
`siNsf`	4	Number of the consecutive NB-IoT downlink subframes that is used to transmit one SI message.	[36.331] Section 5.2.1.2a
`siIndex`	4	The order of entry in the list of SI messages configured by `schedulingInfoList` in `SystemInformationBlockType1-NB`.	[36.331] Section 5.2.3a
`siWindowLengthIndex`	4	Index of common SI scheduling window length for all SIs: 0 – ms160; 1 – ms320; 2 – ms480; 3 – ms640; 4 – ms960; 5 – ms1280; 6 – ms1600.	[36.331] Section 6.7.2
`siRadioFrameOffset`	4	Offset in number of radio frames to calculate the start of the SI window: {1, 2, ..., 15} 0 – indicates that this parameter is not signaled by higher layers. If absent, the UE does not use any offset.	[36.331] Section 6.7.2
`schedulingInfoSiIndex`	3	Indicates the transport block size in number of bits used to broadcast the SI message: 0 – b56; 1 – b120; 2 – b208; 3 – b256; 4 – b328; 5 – b440; 6 – b552; 7 – b680	[36.331] Section 6.7.2
Reserved	1	Reserved	N/A
`nRNTI`	16	Radio network temporary identifier: {0, 1...65535}	[36.321] Section 7.1 Table 7.1-2

4.4.2 PHY_TXSTART Confirmation

This primitive, detailed in Table 25, is issued by the PHY to confirm reception of `PHY_TXSTART.request` primitive and provides the command execution status to the L2. If the PHY is not able to process the `PHY_TXSTART.request` primitive, it returns an error code (`ERRORCODE`) to indicate the issue. The `hyperFrameNumber`, `frameNumber`, and `subFrameNumber` fields are used by the TX state machine to trace events regarding the given subframe.

Note: This message is no longer supported during real-time execution. It is supported only in the c-reference code used to do initial host code integration debugging.

Table 25. PHY_TXSTART Confirmation

Parameter	32-bit Word #	Bits	Length, Bits	Description
`phyEntityId`	0	31:24	8	PHY entity ID (Instance)
`msgType`	0	23:16	8	Message type = 2
`status`	0	15:8	8	Status 0 = Success Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)
`subFrameNum`	0	7:0	8	Subframe number from `TXVECTOR`
`hyperFrameNumber`	1	31:16	16	Hyper frame number
`frameNumber`	1	0:15	16	Frame number

4.4.3 PHY_TXSTART Indication

This primitive is issued by the PHY to indicate to L2 that a downlink subframe has just started. It is sent once per subframe, so it allows synchronization between the PHY and the L2 on a subframe basis. In the real-time execution case, this primitive is used as a TTI delimiter for both TX and RX. Table 26 shows the `PHY_TXSTART` indication details.

Table 26. PHY_TXSTART Indication

Parameter	32-bit Word #	Bits	Length, Bits	Description
`phyEntityId`	0	31:24	8	PHY entity ID (Instance)
`msgType`	0	23:16	8	Message type = 3
`status`	0	15:8	8	Status 0 = Success Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)

Parameter	32-bit Word #	Bits	Length, Bits	Description
subFrameNum	0	7:0	8	Subframe number from TXVECTOR
frameNumber	1	31:16	16	Initial frame number (12 LSBs are used) System frame number
hyperFrameNumber	1	15:0	16	Hyper frame number

4.4.4 PHY_RXSTART Request

This primitive, described in Table 27 is issued by L2 to request the eNB PHY to start the reception of the next uplink subframe. The RXVECTOR carries the control information required by the PHY to receive and decode the UL TBS. After receiving this primitive, the PHY sends a confirmation primitive and, at the proper moment, begins to receive and decode UL symbols. The RXVECTOR corresponds to the Uplink Scheduling Information sent to the remote UEs n-4 subframes earlier. It contains only the UL-Scheduling Information parameters necessary for the PHY to perform the correct receive actions.

Note: For the first subframes, an RXVECTOR containing zero channels is expected.

The eNB version of the RXVECTOR is formatted as specified in Table 28.

Table 29 shows the NPUSCH Format1 configuration parameters for the structure of tNpuschFmt1Configuration, and Table 30 shows the NPUSCH Format2 configuration parameters for the structure of tNpuschFmt2Configuration.

Table 27. PHY_RXSTART Request

Parameter	32-bit Word #	Bits	Size, bits	Description
phyEntityId	0	31:24	8	PHY entity ID (Instance)
msgType	0	23:16	8	Message type = 7
msgSpecific	0	15:0	16	PHY_RXSTART request payload length (ULSUBFRDESC structure)
ULSUBFRDESC structure	N/A	N/A	N/A	PHY_RXSTART parameters. (Refer to Table 28.)

Table 28. UL Subframe Descriptor Parameters—Structure of ULSUBFRDESC

Parameter	Length, Bits	Description, Value	Reference
hyperFrameNumber	16	NB-IoT uplink hyper radio frame number: {0…1023}, 10 LSB bits are actually used with the 6 MSBs zero filled.	[36.331] Section 6.7.2
frameNumber	16	NB-IoT uplink radio frame number: {0…1023}, 10 LSB bits are actually used with the 6 MSBs zero filled.	[36.331] Section 6.2.2
subFrameNumber	8	NB-IoT uplink subframe number: {0…9}, 4 LSB bits are actually used with the 4 MSBs zero filled.	[36.331] Section 6.2.2
nprachIndication	4	Flag to indicate the NPRACH in this subframe: 0 – Indicates no NPRACH opportunity in the subframe. 1 – Indicates one NPRACH opportunity in the subframe, and this NPRACH opportunity uses the first configuration. 2 – Indicates one NPRACH opportunity is in the subframe, and this NPRACH opportunity uses the second configuration. 3 – Indicates one NPRACH opportunity is in the subframe, and this NPRACH opportunity uses the third configuration. 4 ~ 15 – Reserved.	N/A
nprachState	4	State of NPRACH: 0 – Begin at this subframe. 1 – Continue at this subframe. 2 – Halt in order to insert the gaps at this subframe. 3 – End at this subframe. 4 ~ 15 – Reserved.	N/A
npuschFmt1Num	8	Number of NPUSCH format1 in this subframe: {0, 1, 2, …, 47, 48}	N/A

Parameter	Length, Bits	Description, Value	Reference
npuschFmt2Num	8	Number of NPUSCH format2 in this subframe: {0, 1, 2, ..., 8}	N/A
offsetNpuschFmt2Configuration	32	Offset from the beginning of the structure to the element npuschFmt2ConfigurationArray. Since this entry is optional, a value of zero means that this entry is absent for this subframe.	N/A
npuschFmt1ConfigurationArray (npuschFmt1Num structures of tNpuschFmt1Configuration)	npuschFmt1Num X2X32	Refer to Table 29 for more details.	N/A
npuschFmt2ConfigurationArray (npuschFmt2Num structures of tNpuschFmt2Configuration)	npuschFmt2Num X1X32	Refer to Table 30 for more details.	N/A

Table 29. NPUSCH Format1 Configuration Parameters—Structure of tNpuschFmt1Configuration

Parameter	Length, Bits	Description, Value	Reference
nRNTI	16	Radio network temporary identifier for NPUSCH format1: {0, 1...65535}	[36.321] Section 7.1 Table 7.1-2
npuschFmt1State	4	State of NPUSCH format1: 0 – Begin at this subframe. 1 – Continue at this subframe. 2 – Halt in order to insert the gaps at this subframe. 3 – End at this subframe.	N/A

Parameter	Length, Bits	Description, Value	Reference
		4 ~ 15 – Reserved.	
npuschFmt1Msg3Flag	1	Flag to indicate whether this NPUSCH format1 transmits RACH Msg3: 0 – No 1 – Yes	[36.213] Section 16.3.3
subcarrierSpace	1	NB-IoT uplink subcarrier spacing: 0 – 3.75 kHz (Currently, this is the only supported option.) 1 – 15 kHz	[36.213] Section 10.1.2.1
subcarrierIndicationIndex	6	NB-IoT NPUSCH subcarrier indication index that will determine the subcarriers that are used to carry the UL-SCH: For NPUSCH format1 transmission with subcarrier spacing 3.75 kHz, {0, 1…47}, and nsc = Isc. So the number of consecutive subcarriers in an UL resource unit is 1 and the number of consecutive slots in an UL resource unit is 16, according to [36.211] Table 10.1.2.3-1. For NPUSCH format1 transmission with subcarrier spacing 15 kHz, {0, 1…18}, and nsc is determined according to [36.213] Table 16.5.1.1-1. So the number of consecutive subcarriers in an UL resource unit is 1 when Isc < 12, 3 when 12 <= Isc < 16, 6 when 16 <= Isc < 18, and 12 when Isc = 18 and the number of consecutive slots in an UL resource unit is 16 when Isc < 12, 8 when 12 <= Isc < 16, 4 when 16 <= Isc < 18, and 2 when Isc = 18 according to [36.211] Table 10.1.2.3-1.	[36.211] Section 10.1.2.3 Table 10.1.2.3-1 [36.213] Section 16.5.1.1 Table 16.5.1.1-1
modScheme	3	Modulation scheme: 0 – pi/2 BPSK 1 – pi/2 QPSK 2 – QPSK 3 ~ 15 – Reserved	[36.211] Section 10.1.3.2-1

Parameter	Length, Bits	Description, Value	Reference
Reserved1	1	Reserved	N/A
modulationAndCodingSchemeIndex	4	Modulation and coding scheme index: For NPUSCH format1 carrying RACH Msg3 at the first time, {0, 1, 2}, which determines the modulation scheme, and the number of RUs and TBS according to [36.213] Table 16.3.3-1. {3…15} is reserved For NPUSCH format1 carrying other UL-SCH, {0, 1…10}, which determines the modulation scheme and the TBS according to [36.213] Table 16.5.1.2.1-1. {11…15} is reserved.	[36.213] Section 16.3.3 Table 16.3.3-1 [36.213] Section 16.5.1.2.1 Table 16.5.1.2.1-1
resourceAssignmentIndex	3	Resource assignment index: This parameter is only valid for NPUSCH format1 carrying non-RACH-Msg3 at the first time, {0, 1…7}, which determines a number of resource units according to [36.213] Table 16.5.1.1-2.	[36.213] Section 16.5.1.1 Table 16.5.1.1-2
repetitionNumberIndex	3	Repetition number index: This parameter is only valid for NPUSCH format1 carrying UL-SCH, {0, 1…7}, which determines a repetition number for NPUSCH format1 carrying UL-SCH according to [36.213] Table 16.5.1.1-3.	[36.213] Section 16.5.1.1 Table 16.5.1.1-3
schedulingDelayIndex	2	Scheduling delay index: {0, 1, …, 3} This parameter is only valid for NPDCCH with DCI format N0, which determines the first scheduling delay value k0 according to [36.213]] Table 16.5.1-1. For the scheduling of RACH Msg3 at the first time, it also follow the [36.213] Table 16.5.1-1 except for k0 = 13 for IDelay = 0.	[36.213] Section 16.5.1 Table 16.5.1-1
redundancyVersionIndex	1	Redundancy version index: {0, 1}	[36.212] Section 6.4.3.1
newDataIndicator	1	[36.212]	[36.212]

Parameter	Length, Bits	Description, Value	Reference
Reserved2	2	Reserved	N/A
`groupHoppingDisable`	1	Flag to indicate to disable the sequence group hopping for a certain UE: {0, 1}	[36.211] Section 10.1.4.1.3
`groupAssignmentNPUSCH`	5	Group assignment for NPUSCH: {0, 1, ..., 29}	[36.211] Section 10.1.4.1.3
`npuschAllSymbols`	1	Flag to indicate whether the UE uses all NB-IoT symbols for NPUSCH: 0 – The UE punctures the NPUSCH transmissions in the symbols for NPUSCH transmission in the symbols that collides with SRS. 1 – The UE uses all of the NB-IoT symbols for NPUSCH transmissions.	[36.331] Section 6.7.2
`baseSequence`	5	The base sequence of DMRS sequence in a cell for the multitone transmission when group hopping is not enabled: {0, 1, ..., 29} 31 – Indicates that this parameter is not signaled by higher layers. If absent, it is given by NB-IoT cell ID mod 12 for 3-tones; it is given by NB-IoT cell ID mod 14 for 6-tones; it is given by NB-IoT cell ID mod 30 for 12-tones.	[36.211] Section 10.1.4.1.2
`cycleShiftIndex`	2	Define the cyclic shift for multitone transmission: 0 – 0 1 – 2pi/6 2 – 4pi/6 3 – 8pi/6	[36.211] Section 10.1.4.1.2 Table 10.1.4.1.2-3
Reserved3	2	Reserved	N/A

Table 30. NPUSCH Format2 Configuration Parameters—Structure of tNpuschFmt2Configuration

Parameter	Length, Bits	Description, Value	Reference
nRNTI	16	Radio network temporary identifier(RNTI) for NPUSCH format2: {0, 1…65535}	[36.321] Section 7.1 Table 7.1-2
npuschFmt2State	4	State of NPUSCH format2: 0 – Begin at this subframe 1 – Continue at this subframe 2 – Halt in order to insert the gaps at this subframe 3 – End at this subframe 4 ~ 15 – Reserved	N/A
subcarrierSpace	1	NB-IoT uplink subcarrier spacing: 0 – 3.75 kHz (Currently, this is the only supported option.) 1 – 15 kHz	[36.211] Section 10.1.2.3 Table 10.1.2.3-1
subcarrierIndicationIndex	4	NB-IoT NPUSCH subcarrier indication index that will determine the subcarriers that are used to carry the UCI: For NPUSCH format2 transmission with subcarrier spacing 3.75 kHz, {0, 1…15}, nsc and k0 are determined according to [36.213] Table 16.4.2-1. So, the number of consecutive subcarriers in an UL resource unit is 1, and the number of consecutive slots in an UL resource unit is 4, according to [36.211] Table 10.1.2.3-1. For NPUSCH format2 transmission with subcarrier spacing 15 kHz, {0, 1…15}, nsc and k0 are determined according to [36.213] Table 16.4.2-2. So, the number of consecutive subcarriers in an UL resource unit is 1, and the number of consecutive slots in an UL resource unit is 4, according to [36.211] Table 10.1.2.3-1.	[36.211] Section 10.1.2.3 Table 10.1.2.3-1 [36.213] Section 16.4.2-1 Table 16.4.2 [36.213] Section 16.4.2 Table 16.4.2-2

Parameter	Length, Bits	Description, Value	Reference
ackNackNumRepetitionsIndex	3	Number of repetitions for the ACK/NACK resource unit carrying HARQ response: 0 – 1; 1 – 2; 2 – 4; 3 – 8; 4 – 16; 5 – 32; 6 – 64; 7 – 128;	[36.213] Section 16.4.2 [36.331] Section 6.7.2
Reserved	4	Reserved	N/A

4.4.5 PHY_RXSTART Confirmation

This primitive, detailed in Table 31, is generated by the PHY after it has received and processed a complete `PHY_RXSTART.request` primitive. It provides a confirmation of the reception of the request for the command execution status to the L2. If the execution is not successful, the PHY returns an error code (`ERRORCODE`) to indicate the issue. The subframe number field is typically used by the RX state machine to trace events regarding the given subframe.

Note: This message is not issued in the real-time execution of the PHY. It is available for c reference code-based L2 – PHY integration debugging.

Table 31. PHY_RXSTART Confirmation

Parameter	32-bit Word #	Bits	Length, bits	Description
phyEntityId	0	31:24	8	PHY entity ID (Instance)
msgType	0	23:16	8	Message type = 8
status	0	15:8	8	Status 0 = Success Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)
subFrameNum	0	7:0	8	Subframe number from `RXVECTOR`

4.4.6 PHY_RXSTART Indication

This primitive, shown in Table 32, is issued by the PHY to indicate to the MAC the actual start of the UL subframe. The PHY generates this primitive in normal operation mode (after receiving the first `PHY_RXSTART.request` from the L2). The subframe number field is used by the RX state machine to trace events regarding the given subframe. This message is not issued in the real-time execution of the PHY, because the `RXSTART` and `TXSTART` are synchronous and `TXSTART` indication is used as a subframe indication.

Table 32. PHY_RXSTART Indication

Parameter	32-bit Word #	Bits	Length, Bits	Description
`phyEntityId`	0	31:24	8	PHY entity ID (Instance)
`msgType`	0	23:16	8	Message type = 9
`status`	0	15:8	8	Status 0 = Success Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)
`subFrameNum`	0	7:0	8	Subframe number from `RXVECTOR`

4.4.7 Downlink Data

4.4.7.1 PHY_TXSDU Request

This primitive, described in Table 33, is issued by L2 to transfer a transport block (TB) to the PHY, which is uniquely identified by its RNTI using the parameters associated to this channel from the `TXVECTOR`. If there is a difference between the number of bits specified in the `TXVECTOR` for this channel and the number contained in the TB after the PHY performs the coding, modulation, and redundancy check bits addition specified in the `TXVECTOR`, then an `ERRCODE` is given on a per `PHY_TXSDU` basis with a `PHY_TXSDU` confirmation, if enabled.

Note: This error checking is not supported in the real-time execution. It is available for the c-reference code for early host integration debugging.

For all payloads, the convention used is that the most significant bit in the TXSDU payload corresponds to the first in time, or in 3GPP notation a_0 (the first bit to be processed by the PHY).

Table 34 provides details about the downlink PDU request configuration parameters for the structure of `tDlPduReq`.

Table 33. PHY_TXSDU Request Parameters—Structure of TXSDUREQ

Parameter	32-bit Word #	Bits	Size, Bits	Description
`phyEntityId`	0	31:24	8	PHY entity ID (Instance)
`msgType`	0	23:16	8	Message type = 4
`msgLength`	0	15:0	16	`PHY_TXSDU` request payload length (`TXSDUREQ` structure) excluding the first 32-bit word that is used as the NB-IoT PHY SAP message header
`frameNumber`	1	31:16	16	NB-IoT downlink radio frame number: {0...1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.
`hyperFrameNumber`	1	15:0	16	NB-IoT downlink hyper radio frame number: {0...1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.
`numOfPdus`	2	31:16	16	Number of PDUs that will begin to transmit at this subframe: {1, 2} for NPDCCH {1} for NPBCH or NPDSCH
`nbDlChannelType`	2	15:8	8	NB-IoT downlink channel type: 0 – NPSS, only valid when `subFrameNumber` is 5 1 – NSSS, only valid when `subFrameNumber` is 9 and `frameNumber` mod 2 = 0 2 – NPBCH, only valid when `subFrameNumber` is 0 and `frameNumber` mod 64 = 0

Parameter	32-bit Word #	Bits	Size, Bits	Description
				3 – NPDCCH 4 – NPDSCH carrying other content 5 – NPDSCH carrying SIB1-NB 6 – NPDSCH carrying SIBx-NB except for SIB1-NB 7 – NPDSCH carrying paging 8 ~ 15 – Reserved 4 LSB bits are actually used with the 4 MSB's zero filled. **Note:** NB-IoT TxSDU is only valid for NPBCH, NPDCCH, or NPDSCH.
`subFrameNumber`	2	7:0	8	NB-IoT downlink subframe number: {0...9}, 4 LSB bits are actually used with the 4 MSB's zero filled.
`dlPduReqArray` (`numOfPdus` structures of `tDlPduReq`)	3 ~ `numOfPdus` x 2 + 2	N/A	N/A	Refer to Table 34 for more details

Table 34. Downlink PDU Request Configuration Parameters—Structure of tDlPduReq

Parameter	Length, Bits	Description, Value	Reference
`nRNTIOfThisPdu`	16	Radio network temporary identifier for NPUSCH format2: {0, 1...65535}	[36.321] Section 7.1 Table 7.1-2
`numBitsOfThisPdu`	16	Number of bits for the transmission block for this downlink channel beginning to transmit at the subframe	N/A
`offsetOfThisPdu`	32	Offset from the beginning of the structure of TXSDUREQ, excluding the first 32-bit word (SAP message header) to the real PDU payload bit stream	N/A

4.4.8 Uplink Data and Status Indications

4.4.8.1 PHY_RXSDU Indication

The primitive is generated by the PHY to send the contents of a received PHY SDU from the PHY to the L2. The PHY generates this primitive in normal operation mode (after first receiving a `PHY_RXSTART.request` from the L2). There are as many ulPduReports in the `PHY_RXSDU.indication` primitive issued per subframe as unique nRNTIs are specified in the `RXVECTOR`. The format for the `PHY_RXSDU.indication` is shown in Table 35. Table 36 provides details about the uplink PDU report configuration parameters for the structure of `tUlPduReport`.

Table 35. PHY_RXSDU Indication Parameters—Structure of RXSDUIND

Parameter	32-bit Word #	Bits	Size, Bits	Description
`phyEntityId`	0	31:24	8	PHY entity ID (Instance)
`msgType`	0	23:16	8	Message type = 10
`msgLength`	0	15:0	16	`PHY_RXSDU` request payload length (`RXSDUIND` structure) excluding the first 32-bit word that is used as the NB-IoT PHY SAP message header
Reserved	1	31:16	16	Reserved
`numOfPdus`	1	15:0	16	Number of PDUs for NPUSCH format1 that will be reported to higher layer in this subframe: {0, 1, 2, ..., 47, 48}
`ulPduReportArray` (`numOfPdus` structures of tUlPduReport)	2 ~ `numOfPdus` x 5 + 1	N/A	N/A	Refer to Table 36 for more details.

Table 36. Uplink PDU Report Configuration Parameters—Structure of tUlPduReport

Parameter	Length, Bits	Description, Value	Reference
hyperFrameNumber	16	NB-IoT uplink hyper radio frame number: {0…1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.	[36.331] Section 6.7.2
frameNumber	16	NB-IoT uplink radio frame number: {0…1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.	[36.331] Section 6.2.2
subFrameNumber	8	NB-IoT uplink subframe number: {0…9}, 4 LSB bits are actually used with the 4 MSB's zero filled.	[36.331] Section 6.2.2
statusOfThisPdu	8	Status: 0 = Success (valid data) Otherwise ERRCODE (refer to Section 4.2 Error Field Coding)	N/A
nRNTIOfThisPdu	16	Radio network temporary identifier for NPUSCH format1: {0, 1…65535}	[36.321] Section 7.1 Table 7.1-2
timingOfThisPdu	8	Timing advance information in two's complement notation. Value is in terms of TA (16 x Ts @ 30.72 MSa/s). Range is 0 to 63 with a value of 31, which means no change in UE timing is required.	[36.213] Section 16.1.2
snrOfThisPdu	8	SNR for this PDU in steps of 0.5 dB: [0,255] which corresponds to SNR of -64 to +63.5 dB	N/A
rssiDbfsOfThisPdu	16	NPUSCH Fmt2 RSSI in dBFS	N/A
numBitsOfThisPdu	16	Number of bits for the received block for this uplink channel NPUSCH format2	N/A
Reserved	16	Reserved	N/A

Parameter	Length, Bits	Description, Value	Reference
offsetOfThisPdu	32	Offset from the beginning of the structure of RXSDUREQ excluding the first 32-bit word (SAP message header) to the real PDU payload bit stream.	N/A

4.4.8.2 PHY_RXSTATUS Indication

The RXSTATUS.indication is used to report different detection, measurement, and customer-specific information computed or collected by the BS receiver PHY. Table 37, Table 38, and Table 39 provide more information about the RXSTATUS.indication parameters.

Table 37. PHY_RXSTATUS Indication Parameters—Structure of RXSTATUSIND

Parameter	32-bit Word #	Bits	Size, Bits	Description
phyEntityId	0	31:24	8	PHY entity ID (Instance)
msgType	0	23:16	8	Message type = 14
msgLength	0	15:0	16	PHY_RXSTATUS indication payload length (RXSTATUSIND structure) excluding the first 32-bit word used as the NB-IoT PHY SAP message header
Reserved	1	31:16	16	Reserved
numOfAckNack	1	15:8	8	Number of ACK/NACK for NPUSCH format2 that will be reported to a higher layer in the subframe. {0, 1, ..., 6, 7, 8}
numOfRach	1	7:0	8	Number of detected PRACH preambles that will be reported to a higher layer in the subframe. {0, 1,...47, 48}

Parameter	32-bit Word #	Bits	Size, Bits	Description
`offsetOfHarq`	2	31:0	32	Offset from the beginning of the structure of `RXSTATUSIND`, excluding the first 32-bit word (SAP message header) to the element `ulHarqReportArray`. Since this entry is optional, a value of zero means the entry is absent for the subframe
`ulRachReportArray` (`numOfRach` structures of `tUlRachReport`)	3 ~ `numOfRach` x 3 + 2	N/A	N/A	Refer to Table 38 for more details
`ulHarqReportArray` (`numOfAckNack` structures of `tUlHarqReport`)	`numOfRach` x 3 + 3 ~ (`numOfAckNack` x 4) + (`numOfRach` x 3 + 2)	N/A	N/A	Refer to Table 39 for more details

Table 38. Uplink RACH Report Configuration Parameters—Structure of tUlRachReport

Parameter	Length, Bits	Description, Value	Reference
`hyperFrameNumber`	16	NB-IoT uplink hyper radio frame number: {0...1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.	[36.331] Section 6.7.2
`frameNumber`	16	NB-IoT uplink radio frame number: {0...1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.	[36.331] Section 6.7.2
`subFrameNumber`	8	NB-IoT uplink subframe number: {0...9}, 4 LSB bits are actually used with the 4 MSB's zero filled.	[36.331] Section 6.7.2

Parameter	Length, Bits	Description, Value	Reference
preambleIndexDetectedOfThisRach	8	Preamble index detected: {0, 1…47}	[36.213] Section 16.3.3
Reserved	16	Reserved	N/A
timingAdvOfThisRach	16	Timing advance in terms of TA (16*Ts) complement: {0…1282}	[36.213] Section 16.1.2
rssiDbfsOfThisRach	16	RACH RSSI in dBFS	N/A

Table 39. Uplink HARQ Report Configuration Parameters—Structure of tUlHarqReport

Parameter	Length, Bits	Description, Value	Reference
hyperFrameNumber	16	NB-IoT uplink hyper radio frame number: {0…1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.	[36.331] Section 6.7.2
frameNumber	16	NB-IoT uplink radio frame number: {0…1023}, 10 LSB bits are actually used with the 6 MSB's zero filled.	[36.331] Section 6.7.2
subFrameNumber	8	NB-IoT uplink subframe number: {0…9}, 4 LSB bits are actually used with the 4 MSB's zero filled.	[36.331] Section 6.7.2
Reserved1	8	Reserved	N/A
nRNTIOfThisHarq	16	Radio network temporary identifier for this ACK/NACK: {0, 1…65535}	[36.321] Section 7.1 Table 7.1-2
confOfThisHarq	8	Confidence metric for this ACK/NACK	N/A
valueOfThisHarq	8	Value of ACK/NACK: 0 – NACK 1 – ACK	[36.212] Section 6.3.3 Table 6.3.3.1

Parameter	Length, Bits	Description, Value	Reference
`timingAdvOfThisHarq`	8	Timing advance information in two's complement notation. Value is in terms of TA (16 x Ts). Range is 0 to 63 with a value of 31, which means no change in UE timing is required.	N/A
`snrOfThisHarq`	8	SNR for this HARQ in steps of 0.5 dB: [0,255] which corresponds to SNR of -64 to +63.5 dB	N/A
`rssiDbfsOfThisHarq`	16	NPUSCH Fmt2 RSSI in dBFS	N/A
Reserved2	16	Reserved	N/A

4.5 Error Indication

4.5.1 PHY_ERROR _IND

The error indication message, described in Table 40, issues from the PHY when APIs are delivered late by L2/L3 or under other conditions, as stated in Table 7.

Table 40. PHY_ERROR.Indication

Parameter	32-bit Word #	Bits	Length, Bits	Description
`phyEntityId`	0	31:24	8	PHY entity ID (Instance)
`msgType`	0	23:16	8	Message type = 25
`status`	0	15:8	8	Error code from Section 4.2 Error Field Coding
`subFrameNum`	0	7:0	8	Subframe number from TXVECTOR
`frameNumber`	1	31:16	16	System frame number
`hyperFrameNumber`	1	15:0	16	Hyper frame number

§