R E F E R E N C E G U I D E

LibIC-5305

Wireless Communication Protocol

ISM Band 868-870MHz

2011 Jan

Y-Lynx
Rue de Galilée15
CH - 1400 YVERDON
SWITZERLAND

Phone: +41 24 423 92 05
Email: info@y-lynx.com
Web site: http://www.y-lynx.com

Contents

IMPORTANT NOTICE

Documentation updates
Since Y-Lynx products are constantly evolving to meet customer needs, some technical information may differ from those described in this document. Please refer to our web site at www.y-lynx.com to obtain the latest documentation available.

Recommended reading
Other useful documents can be found on our web site www.y-lynx.com

Communication Controller version
Communication Controller software revision: rev1.05

1. GENERAL DESCRIPTION

The LibIC5305 is Y-Lynx's highest performances Radio Communication Controller used in the YLX-TRM8053-xxx-05. Based on an innovative and powerful protocol, the LibIC5305 enables collocated clients and networks operation.

The LibIC5305 radio communication protocol embeds flexible settings to develop complex low power wireless sensor network.

1.1 Features

Protocol and Networking

- Transparent operation
- Point to point, point to multi-point and peer to peer networks.
- API commands to control packet routing
- Embedded protocol
    - Selectable number of hops
    - Dynamic frequencies allocation
- Retries and acknowledgement
- Data Encryption
- Digital RSSI output
- SPI or UART Communication interface
- Hardware protocol Status Tracking
- Safety Mode (Default settings available from ROM memory)
- Two generic input / output

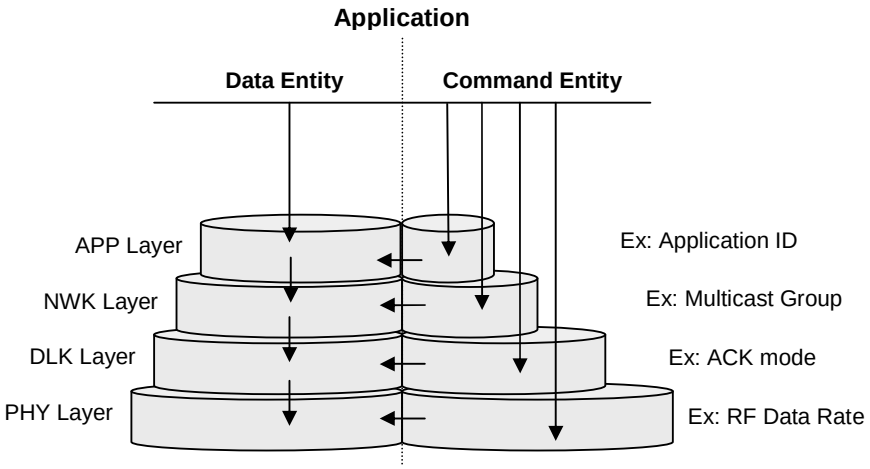
Main Communication Settings

- 3 levels of Address Header
- Packets, semi-packets and not packets data transfer
- Acknowledge mode in the same channel
- Broadcast mode on 7 different groups
- Retry on next channel or on next cycle
- Network information available on the received frame (Sender, Channel, RSSI, …)
- Possibility to assign a fix channel for the transmission
- Flexible frequency table for FHSS protocol
- Possibility to map internal signals on General Purpose I/Os
- Advance configuration available

2 MODEM CONFIGURATION

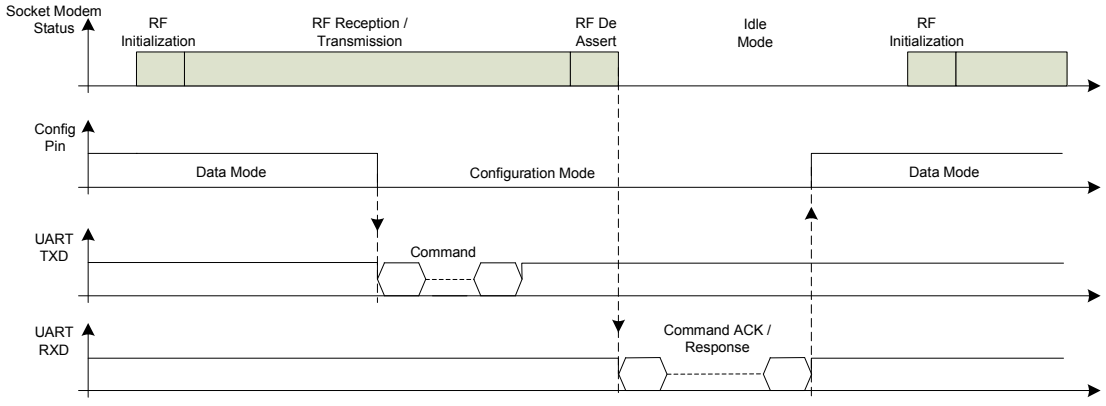
The modem can be configured through four classes of commands and can be related to the OSI Stack.

- APP: Application Layer
- NWK: Network Layer
- DLK: Data Link Layer
- PHY: Physical Layer



2.1 Command Format

To set or read parameters, the modem must first enter “Command Mode” (state in which incoming characters are interpreted as commands). The command can be sent at any time to the radio modem but the response of this command is sent back to the host only during the Idle Mode.



For modified parameter values to persist in the radio modem registry, changes must be saved in Flash memory using SAVE_PARAMETERS Command. Otherwise, parameters are restored to previously saved values when the radio is powered off and then on again.

2.1.1 To Enter in Command Mode

To enter in Command mode two different ways are possible; through the CONFIG/DEFAULT pin or through a UART “Break” condition.

2.1.1.1 Through CONFIG/DEFAULT Pin

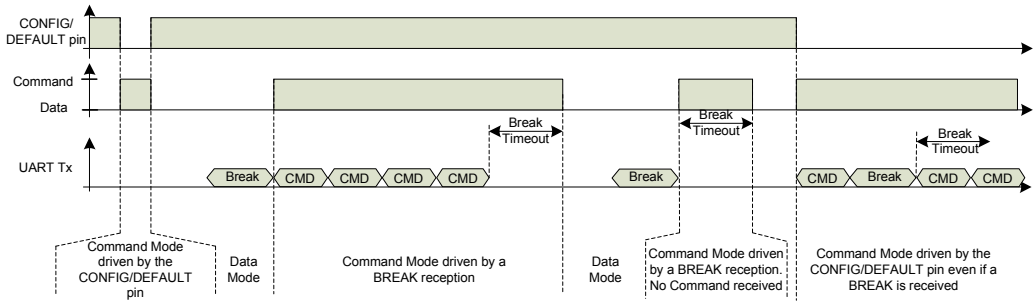
The CONFIG/DEFAULT pin (pin 10) must be driven low to enter in Command Mode and during all the configuration phase.

2.1.1.2 Through a “Break” condition

It is possible to enter in Command mode by sending a “Break” condition on the UART. A break condition is a period of 10 or more low bits received on the UART after a missing stop bit. For example, send a “0x00” at 1200bps to achieve a “Break” condition whatever the UART settings of the radio modem.

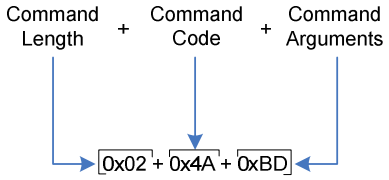
Related commands to the “Break” condition:
- CMD_SET_HOST_BREAK_TIMEOUT
- CMD_GET_HOST_BREAK_TIMEOUT
- CMD_EXIT_HOST_CONFIG_MODE

For more information, please refer to related commands descriptions.



2.1.2 To Send Commands

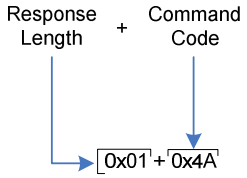
Send commands and parameters using the syntax shown below:



The preceding example would change the RF destination address to “BD” (#189).

2.1.3 Command Acknowledgement

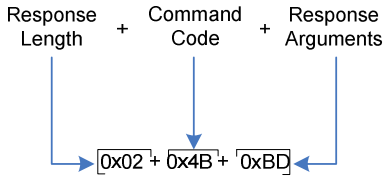
Each SET command sent to the Y-Lynx radio modem is acknowledged by receiving the command code as shown below:



The preceding example confirms the reception by the radio modem of 0x4A Command set (RF destination address).

2.1.4 Command Response

GET Commands requests an answer from the radio modem, the received response uses the syntax shown below:



The preceding example sends back the response to a GET RF_DEST_ADDRESS command (0x4B), the argument is the response to the command, 0xBD (#189) is the RF destination address programmed in the radio modem (the last destination address used by the module).

2.1.5 To Exit from Command Mode

2.1.5.1 Through CONFIG/DEFAULT Pin

To exit from the Command Mode, the CONFIG/DEFAULT pin must be driven high. However, a UART “Break” condition can still force the Command Mode.

2.1.5.2 Through a “Break” condition

The modem is set in Command Mode during a specific time and return after in Data Mode if no command is received by the time (each command reloads the timeout counter). This time is programmable through the following commands:

- CMD_SET_HOST_BREAK_TIMEOUT
- CMD_GET_HOST_BREAK_TIMEOUT

So, the modem returns in Data Mode after the Timeout depending on the status of the CONFIG/DEFAULT pin which can force the Command Mode.

The second way to exit from the Command Mode is to use the specific command:

- CMD_EXIT_HOST_CONFIG_MODE

For more information, please refer to related commands descriptions.

2.1.6 Maximum Size of RF message

The size of a RF message depends on the parameters set in the wireless module as channel duration, RF bit rate, size of the overhead and if the message has to be acknowledged. The Communication Controller embedded in the wireless module is able to calculate the maximum RF message length that can be sent in one channel. The command CMD_GET_RF_FRAME_MAX_SIZE (0xCB) gives the actual maximum value of bytes that can be sent by the modem.

The Communication Controller gives also the possibility to fix the maximum RF message length, by using the following command: CMD_SET_USER_RF_FRAME_MAX_SIZE (0xCE).

So, the maximum RF frame used by the wireless module will be the lowest value between the RF_FRAME_MAX_SIZE and the USER_RF_FRAME_MAX_SIZE.

Design Tips:

In Config Mode, the frame sent by the host to the modem is not submitted to FRAME_MAX_SIZE so if a command is too long, a context error can be generated (ERR_CONTEXT) please for more information, refer to the command description guide.

Channel duration [ms]	Maximum size (bytes) of pload accordingly with the RF bit rate and the ACK mode											
	152340		76170		38085		19042		9521		4760	
	Ack		Ack		Ack		Ack		Ack		Ack	
	off	on	off	on	off	on	off	on	off	on	off	on
10	83	26	32	0	0	0	0					
20	128	128	127	93	52	23	16	0	0			
30			128	128	101	71	40	15	9			
40					128	119	64	39	21	0	0	
50						128	87	62	33	10	6	
60							111	86	45	22	12	
70							128	110	57	34	18	0
80								128	69	46	24	2
90									81	58	30	8
100									93	70	36	14
110									105	82	42	20
120									117	94	48	26
130									128	106	54	32
140										118	60	38
150										128	66	44
160											72	50
170											77	56
180											83	62
190											89	68
200											95	74
210											101	80
220											107	86
230											113	92
240											119	98
250											125	103
260											128	109
270												115
280												121
290												127
300												128

2.2 Command Reference

Table 3.1 Physical layer commands

Command Name	Command	Length*	Range	Description
CMD_SET_RF_POWER	0x60	2 bytes	0x00 0x03	Set the output RF power
CMD_GET_RF_POWER	0x61	1 byte		Get the output RF power
CMD_SET_RF_BITRATE	0x62	2 bytes	0x00 0x05	Set the RF data rate
CMD_GET_RF_BITRATE	0x63	1 byte		Get the RF data rate
CMD_SET_RF_ RECEIVER_MODE	0x6A	2 bytes	0x00 0x01	Set the Receiver in sensitivity or linearity mode
CMD_GET_RF_ RECEIVER_ MODE	0x6B	1 byte		Get the Receiver in sensitivity or linearity mode
CMD_SET_ALL_CHANNEL _TYPE	0x78	14 bytes		Set the type of all channels (receiver, transmitter or idle mode)
CMD_GET_ALL_CHANNEL _TYPE	0x79	1 bytes		Get the type of all channels
CMD_SET_CHANNEL_ TYPE	0x3C	3 bytes	0x00 0x03	Set the channel type (receiver, transmitter or idle mode)
CMD_GET_CHANNEL_ TYPE	0x3D	1 byte		Get the channel type

**Length: includes only command size (1 byte) and parameter range (the request parameter for a SET command and the response parameter for the GET command).*

Table 3.2 Data Link layer commands

Command Name	Command	Length	Range	Description
CMD_SET_ CHANNELS_COUNT	0x20	2 bytes	0x01 0x32	Set the number of used channels
CMD_GET_ CHANNELS_COUNT	0x21	1 byte		Get the number of used channels
CMD_SET_ CHANNEL_DURATION	0x22	2 bytes	0x01 0xFF	Set the duration of a channel
CMD_GET_ CHANNEL_DURATION	0x23	1 byte		Get the duration of a channel
CMD_SET_FREQ_ CORRECTION_MODE	0x28	2 bytes	0x00 0x01	Enable or disable the Frequency correction mode
CMD_GET_FREQ_ CORRECTION_MODE	0x29	1 byte		Get the setting of the Frequency correction mode
CMD_SET_RSSI_MODE	0x2A	2 bytes	0x00 0x03	Enable or disable the Receive Signal Strength Indicator on each channel
CMD_GET_RSSI_MODE	0x2B	1 byte		Get the setting of the Receive Signal Strength Indicator
CMD_SET_ CHANNEL_FREQ	0x2C	4 bytes	0x9A70 0x6590	Set the communication frequency per channel
CMD_GET_ CHANNEL_FREQ	0x2D	2 bytes		Get the communication frequency of one channel
CMD_GET_CHANNELS_ MAX_COUNT	0x2F	1 byte	0x32	Get the maximum of available channels
CMD_GET_LAST_ BEACON_INFO	0x31	2 bytes	-	Gives information on the last received Beacon
CMD_GET_LAST_TX_ FRAME_INFO	0x33	2 bytes	-	Gives information on the last transmited frame
CMD_GET_LAST_RX_ FRAME_INFO	0x37	2 bytes	-	Gives information on the last received frame
CMD_SET_RF_ ADDRESS	0x44	2 bytes	0x00 0xFE	Set an address to the device
CMD_GET_RF_ ADDRESS	0x45	1 byte		Get the device address
CMD_SET_RF_ DEST_ADDRESS	0x4A	2 bytes	0x00 0xFF	Set a destination address
CMD_GET_RF_	0x4B	1 byte		Get the destination address

DEST_ADDRESS				
CMD_SET_RF_ ACK_MODE	0x50	2 bytes	0x00 0x01	Enable/disable the ACK Mode
CMD_GET_RF_ ACK_MODE	0x51	1 byte		Get the status of ACK Mode
CMD_SET_RF_ CHECK_MODE	0x52	2 bytes	0x00 0x03	Set the redundancy check mode
CMD_GET_RF_ CHECK_MODE	0x53	1 byte		Get the status of the redundancy check mode
CMD_SET_BEACON_ MODE	0x54	2 bytes	0x00 0x02	Set the device in server, client or beaconless Mode
CMD_GET_BEACON_ MODE	0x55	1 byte		Get the mode of the device (server, client or beaconless)
CMD_SET_BEACON_ PERIOD	0x56	2 bytes	0x01 0x05	Set the number of sequence between two beacon sent by the server on channel “0”
CMD_GET_BEACON_ PERIOD	0x57	1 byte		Get the setting of the beacon period set in the server
CMD_SET_BEACON_ LISTEN_PERIOD	0x58	2 bytes	0x01 0x0A	Set the number of sequence between two beacon received by the client on channel “0”
CMD_GET_BEACON_ LISTEN_PERIOD	0x59	1 byte		Get the setting of the beacon listen period set in the client
CMD_SET_MAX_ BEACON_LOST	0x5A	2 bytes	0x00 0x0A	Set the number of beacon that a client can lost before to be considered as desynchronized (10 maximum)
CMD_GET_MAX_ BEACON_LOST	0x5B	1 byte		Get the number of beacon that the client can lost before to be considered as desynchronized
CMD_SET_RF_RETRY_ COUNT	0x5C	2 bytes	0x00 0x07	Set the number of retry that can be sent for a given RF packet
CMD_GET_RF_RETRY_ COUNT	0x5D	1 byte		Get the number of retry that can be sent for a given RF packet
CMD_SET_MAX_ BEACON_TRACK	0x5E	2 bytes	0x00 0xFF	Set the number of beacon period to try to resynchronized a client (255 maximum)
CMD_GET_MAX_ BEACON_TRACK	0x5F	1 byte		Get the number of beacon period used to resynchronized a client
CMD_SET_ EXTENDED_ PROTOCOL	0x74	2 bytes	0x00 0x01	Activate the extended protocol
CMD_GET_EXTENDED_ PROTOCOL	0x75	1 byte		Get the protocol option
CMD_SET_TX_ CHANNEL	0xC6	2 bytes	0x00 0x31	Used to set the channel restriction for the transmission
CMD_GET_TX_ CHANNEL	0xC7	1 byte		Get the channel restriction for a transmission
CMD_SET_TX_RETRY_ RESTRICTION	0xC8	2 bytes	0x00 0x01	Used to set the retries channel restriction for the transmission
CMD_GET_TX_RETRY_ RESTRICTION	0xC9	1 byte		Get the channel retries restriction for a transmission
RF_FRAME_MAX_SIZE	0xCB	1 byte	0x00 0x80	Get the absolute maximum size of a RF frame
CMD_SET_USER_RF_ FRAME_MAX_SIZE	0xCE	2 bytes	0x00 0x80	Used to set the maximum size of the RF frame defined by the user
CMD_GET_USER_RF_ FRAME_MAX_SIZE	0xCF	1 byte		Used to read the maximum size value of the RF frame defined by the user

Table 3.3 Network layer commands

Command Name	Command	Length	Range	Description
CMD_SET_NWK_ID	0x42	2 bytes	0x00 0x3F	Set an id to the network
CMD_GET_NWK_ID	0x43	1 byte		Get the id of the network
CMD_SET_RF_ MULTICAST	0x4C	2 bytes	0x00 0xFF	Register a node to one or more group
CMD_GET_RF_ MULTICAST	0x4D	1 byte		Get the registered group of the node
CMD_SET_RF_DEST MULTICAST	0x4E	2 bytes	0x00 0xFF	Define the destination multicast
CMD_GET_RF_DEST MULTICAST	0x4F	1 byte		Get the destination multicast

Table 3.4 Application commands

Command Name	Command	Length	Range	Description
CMD_GET_VERSION	0x10	1 byte	-	Get the version of the module
CMD_GET_LIBIC_ VERSION	0x11	1 byte	-	Get the LibIC version
CMD_SET_APPL_ID	0x40	2 bytes	0x00 0xFF	Set an id to the application
CMD_GET_APPL_ID	0x41	1 byte		Get the id of the application
CMD_SET_REMOTE_ CMD_MODE	0x76	2 bytes	0x00 0x01	Configure the module to accept remote command
CMD_GET_REMOTE_ CMD_MODE	0x77	1 byte		Get the command mode setting of the module
CMD_SET_CIPHER_MODE	0xA2	2 bytes	0x00 0x01	Set the module in cipher mode
CMD_GET_CIPHER_MODE	0xA3	1 byte		Get the cipher mode setting
CMD_SET_CIPHER_KEY	0xA4	5 bytes	0x00 0xFF	Set the cipher code (key)
CMD_GET_CIPHER_KEY	0xA5	1 byte		Get the cipher code
CMD_SET_HOST_ BYTE_TIMEOUT	0xC0	2 bytes	0x00 0x23	Configure the timeout between two bytes, used to determine the “end of frame”
CMD_GET_HOST_ BYTE_TIMEOUT	0xC1	1 byte		Get the programmed byte timeout value
CMD_SET_HOST_ BREAK_TIMEOUT	0xC2	2 bytes	0x00 0xFF	Configure the timeout after which a break condition is exited
CMD_GET_HOST_ BREAK_TIMEOUT	0xC3	1 byte		Get the timeout after which a break condition is exited
CMD_SET_HOST_ BAUDRATE	0xC4	2 bytes	0x00 0x07	Configure the baud rate of the UART interface with the host
CMD_GET_HOST_ BAUDRATE	0xC5	1 byte		Get the programmed baud rate
CMD_EXIT_HOST_ CONFIG_MODE	0xCA	1 byte	-	Exit configuration mode
CMD_SET_HOST_UART_ SETTINGS	0xCC	2 bytes	0x00 0x1F	Configure the UART parameters
CMD_GET_HOST_ UART_ SETTINGS	0xCD	1 byte		Get the UART parameters
CMD_SET_HOST_DATA_ MODE	0xD0	2 bytes	TX 0x00 0x3F	Used to set the configuration of the data mode
CMD_GET_HOST_DATA_ MODE	0xD1	1 byte	RX 0x00 0xFF	Used to read the data mode configuration
CMD_SEND_DATA	0xD2	n bytes	-	Send data in configuration mode
IND_RECEIVED_DATA	0xD3	n bytes	-	Used to receive data in configuration mode
CMD_SEND_ECHO_ DATA	0xD4	n bytes	-	Allow to send an RF data frame to a hostless modem which will echo it back
CMD_SET_POWER_ MODE	0xD6	2 bytes	0x00 0x02	Used to set the power mode of the radio modem.

CMD_GET_POWER_ MODE	0xD7	1 byte		Used to read the power mode of the radio modem
CMD_SEND_COMMAND	0xD8	n bytes	-	Send a command to a remote wireless module
IND_RECEIVED_ RESPONSE	0xD9	n bytes	-	Used to receive the response to the sent command from the remote wireless module
CMD_SET_CTS_ THRESHOLD	0xDA	2 bytes	0x00 0xFF	Used to fine tune the CTS signal behavior.
CMD_GET_CTS_ THRESHOLD	0xDB	1 byte		Used to read the settings of CTS threshold
CMD_SET_HOST_DATA_ END_CONDITION	0xDC	2 bytes	0x00 0x03	Used to set the RF frame end condition
CMD_GET_HOST_DATA_ END_CONDITION	0xDD	1 byte		Reads the host data frame end condition
CMD_SET_RESET_MODE	0xDE	2 bytes	0x00 0x03	Determine which configuration has to be loaded after the reset and on which conditions
CMD_GET_RESET_MODE	0xDF	1 byte		Get the configuration of the reset mode
CMD_SET_GP1_SIGNAL	0xE0	2 bytes	0x00 0x04	Map a signal on GP1
CMD_GET_GP1_SIGNAL	0xE1	1 byte		Get the mapping on the GP1
CMD_SET_GP1_VALUE	0xE2	2 bytes	0x00 0x01	Write a signal level on GP1 when configured as output
CMD_GET_GP1_VALUE	0xE3	1 byte		Read the signal level on GP1 when configured as input
CMD_SET_GP2_SIGNAL	0xE4	2 bytes	0x00 0x05	Map a signal on GP2
CMD_GET_GP2_SIGNAL	0xE5	1 byte		Get the mapping on the GP2
CMD_SET_GP2_VALUE	0xE6	2 bytes	0x00 0x01	Write a signal level on GP2 when configured as output
CMD_GET_GP2_VALUE	0xE7	1 byte		Read the signal level on GP2 when configured as input
CMD_SET_USER_DATA	0xF0	5 bytes	-	Allow to save the user version number
CMD_GET_USER_DATA	0xF1	1 byte		Get the user version number
CMD_SAVE_ PARAMETERS	0xF2	2 bytes	0x00 0x08	Save the current configuration in Flash memory
CMD_LOAD_ PARAMETERS	0xF4	2 bytes	0x00 0x08	Load the Flash memory in RAM
CMD_LOAD_DEFAULT_ PARAMETERS	0xF6	2 bytes	0x00 0x08	Load the ROM memory in RAM

Table 3.5 and 3.6 Error command and Error codes

Command Name	Command	Length	Range	Description
IND_ERROR	0x03	2 bytes	-	Sent in response to a request when an error occurred

Error Code Name	Code	Description
ERR_FORMAT_CMD	0x02	The command does not respect the protocol
ERR_INVALID_ARGUMENT	0x03	An argument is invalid
ERR_OUTOFRANGE_ARGUMENT	0x04	An argument is out of range
ERR_TIMEOUT	0x05	Byte timeout occurred before complete command reception
ERR_CONTEXT	0x06	The command can not be executed with the current parameters
ERR_UNDEFINED_CMD	0x07	The module does not know how to interpret the given command

2.3 Command Description

Notes:

Commands in this section are listed **ALPHABETICALLY**.

- CMD_SET_ and CMD_GET_ prefix are not taken into account.
- Command categories are designated between “< >” symbols placed after each command title.

Example: To search CMD_SET_POWER_MODE, go to POWER_MODE

ALL_CHANNELS_TYPE Command (0x78 / 0x79) **<PHY Layer>**

Purpose:
This command is used to attribute a specific mode (transmit, receive or idle) to a specific channel and for all defined channels. This command is often used to set a restriction on a channel, for example to ban a transmission on one specific channel or to force the idle mode (low power mode).

For more information about this command please refer to the module datasheet.

WARNING:
This command is available only if the extended protocol is enabled. EXTENDED_PROTOCOL (0x74 / 0x75)

Command SET: 0x78; Length of the command: 0xD bytes
Command GET: 0x79; Length of the command: 0x01 byte

Parameter range: Byte: 0x00: all channels are in idle mode
to
Byte: 0xFF: all channels can receive and transmit data

Default parameter: 0x0F: all channels can receive and transmit data

Example:
The host set the type of all the channels by sending

	Byte 1	Byte 2	Byte 3	Byte 4	…	Byte 14
From host to the module	0x0D	0x78	Config byte 0	Config byte 1	…	Config byte 12

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x79	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

APPL_ID Command (0x40 / 0x41) **<APP Layer>**

Purpose:
These commands are used to set/read the application identification of a device. The YLX-TRM module uses three address layers. To enable a communication between clients, the APP_ID and the NWK_ID must be identical.

Command SET: 0x40; Length of the command: 0x02 bytes
Command GET: 0x41; Length of the command: 0x01 byte

Parameter range: 0x00: Address 0
to
0xFF: Address 255

Default parameter: 0x01: Application number 1

Example:
The host gets the application ID by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x41	-	-	-

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x41	0x00	-	-

The host set the Application ID to 0x01 by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x40	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x40	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

BEACON_LISTEN_PERIOD Command (0x58 / 0x59) **<DLK Layer>**

Purpose:
This command is used in Client Mode, and set/read the listen period of the beacon. This variable is used to optimize the power consumption and the synchronization times for each Client, as each of them have different constraints. The beacon listen period defines the number of beacons period to listen to. Please refer also to CHANNELS_COUNT and BEACON_PERIOD commands.

Design Tips:
In Frequency Hopping Spread Spectrum protocol, the number of channels is set to 50 per seconds with channel duration of 20ms. If the BEACON_PERIOD is set to 3, the beacon signal will be sent every 3 seconds on channel "0". If the BEACON_LISTEN_PERIOD of a Client is set to 2, the client will synchronized itself every 3x2 sequences, so every 6 seconds (in this example the sequence duration is one second).

Command SET: 0x58; Length of the command: 0x02 bytes
Command GET: 0x59; Length of the command: 0x01 byte

Parameter range: 0x01: The beacon synchronization is done on every beacon signal.
to
0x0A: The beacon synchronization is done on every 10 beacon signals.

Default parameter: 0x01: Beacon synchronization on every beacon signal

Example:
The host gets the "beacon listen period" setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x59	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x59	0x01	-	-

The host set the "beacon listen period" to every 2 beacon signals by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x58	0x02	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x58	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

BEACON_MODE Command (0x54 / 0x55) **<DLK Layer>**

Purpose:
To establish synchronization between transceivers, a beacon is emitted by one module. This radio modem is called Server and is in charge to synchronize the entire network. The purpose of this command is to set/read the Beacon Mode to identify the unique network Server.

It is also possible to synchronize the module on an external signal (the signal has to be mapped on GP2) instead of RF beacon, this mode is called beaconless.

Command SET: 0x54; Length of the command: 0x02 bytes
Command GET: 0x55; Length of the command: 0x01 byte

Parameter range: 0x00: Beacon Mode Client
0x01: Beacon Mode Server
0x02: Beaconless network

Default parameter: 0x00, The module is set in Client Mode

Example:
The host gets the "Beacon Mode" setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x55	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x55	0x00	-	-

The host set the device in Server Mode (0x01) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x54	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x54	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

BEACON_PERIOD Command (0x56 / 0x57) **<DLK Layer>**

Purpose:
To optimize the current consumption but also to reduce the synchronization time, a beacon period is implemented. This beacon period defines the number of sequences between two beacons. The beacon signal is always sent on channel "0", the first channel set in the frequency table. Please refer also to CHANNELS_COUNT command for more information.

Design Tips:
In Frequency Hopping Spread Spectrum protocol, the number of channels is set to 50 per seconds with channel duration of 20ms. If the BEACON_PERIOD is set to 3, the beacon signal will be sent every 3 seconds on channel "0".

Command SET: 0x56; Length of the command: 0x02 bytes
Command GET: 0x57; Length of the command: 0x01 byte

Parameter range: 0x01: Beacon signal is sent at the beginning of every sequence
to
0x05: Beacon signal is sent at the beginning of every 5 sequences

Default parameter: 0x01, A beacon is sent on every sequence

Example:
The host gets the "beacon period" setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x57	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x57	0x01	-	-

The host set the beacon period to every 5 sequence (0x05) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x56	0x05	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x56	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

CHANNELS_COUNT Command (0x20 / 0x21) **<DLK Layer>**

Purpose:
These commands are used to set/read the number of channels used in the network. This number is not dependant of a time base. It's possible to set a number of channels and to use only a part of. By using the CHANNEL_DURATION command, the number of hops is not limited to 1 second.

Design Tips:

A network can be configured in order to achieve 5 hops per second with 50 frequencies. In this case a radio communication on the frequency f1 will be achieved every 10 seconds.

CHANNELS_COUNT = 50
CHANNEL_DURATION = 20 (time slot = 20x10ms)

Command SET: 0x20; Length of the command: 0x02 bytes
Command GET: 0x21; Length of the command: 0x01 byte

Parameter range: 0x01: Only one frequency used in the network (TDMA)
to
0x32: 50 frequencies (FHSS)

Default parameter: 0x32, 50 frequencies

Example:
The host gets the channels count by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x21	-	-	-

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x21	0x32	-	-

The host set the channels count to 10 by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x20	0x0A	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x20	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

CHANNEL_DURATION Command (0x22 / 0x23) **<DLK Layer>**

Purpose:
These commands are used to set/read the time spent per channel. Used with the CHANNELS_COUNT command, it allows defining the network topology as TDMA, FDMA and FHSS. The variable is based on 1 byte with a resolution of 10ms. In addition, the minimum channel duration is fixed to 10ms.

Using a lower CHANNEL_DURATION can reduce the global data throughput. This is due to the channel initialization time (~3.6ms) as well as the frame encapsulation overhead. Both are constant and neither relative to the RF_BITRATE nor CHANNEL_DURATION. Thus multiplying the channels by decreasing their duration reduces the payload effective transmission time.

Design Tips:

The network requests a time slot configuration of 40ms
CHANNEL_DURATION = 4 (time slot = 4x10ms)

Command SET: 0x22; Length of the command: 0x02 bytes
Command GET: 0x23; Length of the command: 0x01 byte

Parameter range: 0x01: Minimum channel duration, 1 x 10ms
to
0xFF: Maximum channel duration 255 x 10ms

Default parameter: 0x02, Duration of 20ms per time slot

Example:
The host gets the channels duration by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x23	-	-	-

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x23	0x02	-	-

The host set the channels duration to 40ms by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x22	0x04	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x22	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

CHANNEL_FREQ Command (0x2C / 0x2D) **<DLK Layer>**

Purpose:
These commands are used to set/read the frequency of a channel. Each channel is defined by a **Channel ID** and a **Frequency**. The **Channel ID** is an identification code. The frequency is calculated on 2 bytes and from the middle of the band. The frequency resolution is fixed with steps of 500Hz.

Design Tips:

869'000'000 Hz = Middle of the band = 0x0000
869'000'500 Hz = Middle of the band + 1 * 500Hz = 0x0001
868'999'500 Hz = Middle of the band – 1 * 500Hz = 0xFFFF

Command SET: 0x2C; Length of the command: 0x04 bytes
Command GET: 0x2D; Length of the command: 0x02 bytes

Parameter range: 0 ≤ **Channel ID** < CHANNELS_MAX_COUNT (50)

For the 869MHz band:
From: 0xF830: Lowest frequency (-1MHz from 869MHz)
0x0000: Middle of the Band
To: 0x07D0: Highest frequency (+1MHz from 869MHz)

Default parameter: See Table in appendix

Example:
The host gets the frequency of the channel 26 (0x1A) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x2D	0x1A	-	-

The radio modem responds
(the default value equals to 869'796'000 Hz = 0x0638 x 500Hz)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x04	0x2D	0x1A	0x06	0x38

The host set the channel 49 (0x31) at the frequency 870'000'000 Hz (0x07D0) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x04	0x2C	0x31	0x07	0xD0

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x2C	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

CIPHER_KEY Command (0xA4 / 0xA5) **<APP Layer>**

Purpose:
These commands are set/read the 32-bits used by for the encryption (cipher mode)

Command SET: 0xA4; Length of the command: 0x05 bytes
Command GET: 0xA5; Length of the command: 0x01 bytes

Parameter range: Key Code: KEY0 - KEY1 - KEY2 - KEY3
From 0x00
To 0xFF

Default parameter: -

Example:
The host set cipher key setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From host to the module	0x05	0xA4	KEY0	KEY1	KEY2	KEY3

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xA4			

The host read the cipher key by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xA5			

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From module to the host	0x05	0xA5	KEY0	KEY1	KEY2	KEY3

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

CIPHER_MODE Command (0xA2 / 0xA3) **<APP Layer>**

Purpose:
These commands are used to enable or disable the 32-bits encryption (cipher mode)

Command SET: 0xA2; Length of the command: 0x04 bytes
Command GET: 0xA3; Length of the command: 0x01 bytes

Parameter range: 0x00: Encryption disabled
0x01: Encryption enabled

Default parameter: 0x00: Encryption disabled

Example:
The host gets cipher mode setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xA3		-	-

The radio modem responds (default value)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xA3	0x00		

The host enables the cipher mode by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xA2	0x01		

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xA2	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

CTS_THRESHOLD Command (0xDA / 0xDB) **<APP Layer>**

Purpose:
The CTS_THRESHOLD parameter is used to fine tune the CTS signal behavior to insure that the host will sample in time. See also HOST_ FRAME_MAX_SIZE command.

Command SET: 0xDA; Length of the command: 0x02 bytes
Command GET: 0xDB; Length of the command: 0x01 byte

Parameter range: 0x00: 0 byte less than HOST_ FRAME_MAX_SIZE
To:
0xFF: 255 bytes less than the HOST_ FRAME_MAX_SIZE

Default parameter: 0x00

Example:
The host gets CTS Threshold value by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xDB	-	-	-

The radio modem responds that the value is 0 byte

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xDA	0x00	-	-

The host set the channel CTS Threshold to 7 bytes value by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xDA	0x07	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xDA	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

CHANNELS_MAX_COUNT Command (0x2F) **<DLK Layer>**

Purpose:
This command is used to read the maximum channels managed by the Y-Lynx radio modem. This number is fixed by the module architecture to 50 channels (0x32).

Design Tips:

The maximum number of available channels can differ from Y-Lynx module to another one. Before to develop a network, in particularly custom architecture, send the GET_CHANNEL_MAX_COUNT to the Y-Lynx module to obtain the maximum number of available channels.

Command GET: 0x2F; Length of the command: 0x01 byte

Parameter range: -

Default parameter: -

Example:
The host gets the Maximum channels available by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x2F	-	-	-

The radio modem responds 50 channels (0x32)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x2F	0x32	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

CHANNEL_TYPE Command (0x3C / 0x3D) **<PHY Layer>**

Purpose:
The channel type command is used to attribute a specific mode (transmit, receive or idle) to a specific channel. This command is often used to set a restriction on a channel, for example to ban a transmission on one specific channel or to force the idle mode (low power mode).

WARNING:
This command is available only if the extended protocol is enabled. EXTENDED_PROTOCOL (0x74 / 0x75)

Command SET: 0x3C; Length of the command: 0x03 bytes
Command GET: 0x3D; Length of the command: 0x02 byte

Parameter range:
0x00: Idle mode
0x01: Receiver mode only
0x02: Transmitter mode only (if data are present)
0x03: Receive and transmit mode

Default parameter: 0x03: Receive and transmit mode

Example:
The channel #4 will be set in receiver mode only, so the module will not be able to send data on this channel. For that the following command should be sent.

The host set the channel #4 in receiver only

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x03	0x3C	0x04	0x01	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x3C	-	-	-

The host gets the type of the channel #4

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x03	0x3D	0x04		-

To module sends the following response (receiver only)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x03	0x3D	0x04	0x01	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

ERROR Indicator (0x03) **<ERROR>**

Purpose:
This message is sent by the radio modem to host in response to a request when an error occurred.

Length of the command: 0x02 bytes

Command: -

Parameter range:
0x02: The command does not respect the protocol
0x03: Argument is invalid
0x04: An argument is out of range
0x05: Byte timeout before end of command (CMD)
0x06: Incompatibility with the current parameters
0x07: The module does not know how to interpret the given command

Default parameter: -

Example:
If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

EXIT_HOST_CONFIG_MODE Command (0xCA) **<APP Layer>**

Purpose:
This command is used to exit from the Command mode without waiting the occurrence of the Break Timeout. The exit from the Command mode is activated only if the CONFIG/DEFAULT pin is set in Data Mode. If not, the modem stays in Command Mode.

Command: 0xCA; Length of the command: 0x01 byte

Parameter range: -

Default parameter: -

Example:
The host exits from the Command Mode by sending (the CONFIG/DEFAULT pin is set in Data Mode)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xCA	-	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xCA		-	-

The host stays in Command Mode (the CONFIG/DEFAULT pin is set in Command Mode) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xCA		-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xCA	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

EXTENDED_PROTOCOL Command (0x74 / 0x75) **<DLL Layer>**

Purpose:
This command is used to enable or disable the extended protocol option. When enabled, more commands are available like remote commands or channel configuration, the compatibility with previous version, before version V1.03, is not possible. When disabled, the protocol is compatible with previous protocol version up to the version V1.02 or later if the extended protocol is disabled

Command SET: 0x74; Length of the command: 0x02 bytes
Command GET: 0x75; Length of the command: 0x01 byte

Parameter range: 0x00: Extended protocol disabled
- Compatible with previous version up to V1.02
- Compatible with the version V1.03 or later if extended protocol is disabled

0x01: Extended protocol enabled.

Default parameter: 0x00: Extended protocol disabled

Example:
The host gets the extended protocol setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x75	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x75	0x00	-	-

The host configure the module to accept the extended protocol by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x74	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x74	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

FREQ_CORRECTION_MODE Command (0x28 / 0x29) **<DLK Layer>**

Purpose:
These commands are used to enable/disable the Frequency Correction. When enabled, the frequency offset is corrected thanks to the beacon reception.

Command SET: 0x28; Length of the command: 0x02 bytes
Command GET: 0x29; Length of the command: 0x01 byte

Parameter range: 0x00: Frequency Correction is not enabled
0x01: Frequency Correction is enabled

Default parameter: 0x01, Frequency Correction is enabled

Example:
The host gets the Frequency Correction Mode setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x29	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x29	0x01	-	-

The host disables the Frequency Correction by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x28	0x00	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x28	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

GP1_SIGNAL Command (0xE0 / 0xE1) **<APP Layer>**

Purpose:
These commands are used to map signals on GP1.

WARNING:
Before to set GP1 or GP2 signal as general purpose output; it is highly recommended to set the signal value first by using the command SET_GP1_VALUE or SET_GP2_VALUE. It insures a known transition from high impedance to high level or to low level.

Command SET: 0xE0; Length of the command: 0x02 bytes
Command GET: 0xE1; Length of the command: 0x01 byte

Parameter range:
0x00: GP1_GPI; General purpose input, the value of the pin is read thanks to the command GET_GP1_VALUE
0x01: GP1_GPO; General purpose output, the value of the pin is written thanks to the command SET_GP1_VALUE
0x02: GP1_TX_PENDING;
=1, when data from the host are ready to be sent
=0, when data in the transmit buffer are not ready to be sent
When two frame are ready to be sent, GP1 goes low "0", when the first frame stored in the transmit buffer has been sent and goes back to "1" to prepare the transmission of the next frame stored in the second transmit buffer.
0x03: HOST_ACTIVITY; active signal when data are transmitted from the radio modem to the host. This signal can be used to control the charge pump of the UART/RS232 converter. This signal is enabled 100µs before to transmit the first byte and disabled 1s after the transfer of the last byte to the host.
0x04: CTS is mapped on GP1

Default parameter: 0x00, General purpose input

Example:
The host gets the GP1 setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xE1	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xE1	0x00	-	-

The host set the GP1 signal as output by sending (cf. SET_GP1_VALUE)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xE0	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xE0	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

GP1_VALUE Command (0xE2 / 0xE3) **<APP Layer>**

Purpose:
These commands are used to read and write values on GP1, when it is configured as General Purpose Input / Output (c.f. GP1_SIGNAL).

Command SET: 0xE2; Length of the command: 0x02 bytes, used if GP1 is mapped as an output

Command GET: 0xE3; Length of the command: 0x01 byte, used if GP1 is mapped as an input

Parameter range: 0x00: The signal GP1 goes to low level if GP1 is set as output
0x01: The signal GP1 goes to high level, if GP1 is set as output

Default parameter: 0x00, Low level

Example as output:
The host set the GP1 signal to high level by sending, (GP1 has been set as output)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xE2	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xE2	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

Example as input:
The host get the level present on GP1 signal by sending, (GP1 has been set as input)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xE3		-	-

To Radio Modem sends the level present on GP1 signal (a high level is present on GP1)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xE3	0x01	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

GP2_SIGNAL Command (0xE4 / 0xE5) **<APP Layer>**

Purpose:
These commands are used to map signals on GP2.

WARNING:

Before to set GP1 or GP2 signal as general purpose output; it is highly recommended to set the signal value first by using the command SET_GP1_VALUE or SET_GP2_VALUE. It insures a known transition from high impedance to high level or to low level.

Command SET: 0xE4; Length of the command: 0x02 bytes
Command GET: 0xE5; Length of the command: 0x01 byte

Parameter range: 0x00: GP2_GPI; General purpose input, the value of the pin is read thanks to the command GET_GP2_VALUE
0x01: GP2_GPO General purpose output, the value of the pin is written thanks to the command SET_GP2_VALUE
0x02: GP2_TX_ACK; gives the status of the last RF acknowledgement.
=1, the last RF communication with a distant modem has been acknowledged.
=0, the last RF communication with a distant modem has not been acknowledged
0x03: GP2_RF_CMD_TX; Indicates if the modem is busy with an RF communication requested by a distant modem (and not by the host).
=1; the modem is busy (See CMD_SEND_ECHO_ DATA)
=0; the modem is free to send host data over RF
0x04: RTS is mapped on GP2
0x05:GP2 is configured as an input to accept an external synchronization signal (SYNC_IN) used in beaconless network.

Default parameter: 0x00, General purpose input

Example:
The host gets the GP2 setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xE5	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xE5	0x00	-	-

The host set the GP2 signal as output by sending (cf. SET_GP2_VALUE)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xE4	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xE4	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

GP2_VALUE Command (0xE6 / 0xE7) **<APP Layer>**

Purpose:
These commands are used to read and write values on GP2, when it is configured as General Purpose Input / Output (c.f. GP2_SIGNAL).

Command SET: 0xE6; Length of the command: 0x02 bytes, used if GP2 is mapped as an output

Command GET: 0xE7; Length of the command: 0x01 byte, used if GP2 is mapped as an input

Parameter range: 0x00: The signal GP2 goes to low level if GP2 is set as output
0x01: The signal GP2 goes to high level if GP2 is set as output

Default parameter: 0x00, Low level

Example as output:
The host set the GP2 signal to high level by sending, (GP2 has been set as output)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xE6	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xE6	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

Example as input:
The host get the level present on GP2 signal by sending, (GP2 has been set as input)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xE7		-	-

To Radio Modem sends the level present on GP2 signal (a high level is present on GP2)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xE7	0x01	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

HOST_BAUDRATE Command (0xC4 / 0xC5) **<APP Layer>**

Purpose:
These commands are used to set/read the communication baud rate of the UART between the radio modem and the host.

WARNING:

The host baud rate is changed after acknowledgement of the command.

Design Tips:

When decreasing the HOST_BAUDRATE, be sure that the current HOST_BYTE_TIMEOUT is not too short regarding the future byte duration.
For example, at 9600 bauds with UART sets with 8,N,1; a byte duration corresponds to (1/9600) * 10 bits, so 1.04ms. In this case if the value of HOST_BYTE_TIMEOUT is equal to or less than 1.0ms, a byte timeout error will occur in command mode and only frames of a unique data byte will be obtained in data mode.

Command SET: 0xC4; Length of the command: 0x02 bytes
Command GET: 0xC5; Length of the command: 0x01 byte

Parameter range:
0x00: 9600 bauds
0x01: 19200 bauds
0x02: 38400 bauds
0x03: 57600 bauds
0x04: 115200 bauds
0x05: 1200 bauds
0x06: 2400 bauds
0x07: 4800 bauds

Default parameter: 0x00, The communication with the host is fixed to 9600 bauds

Example:
The host gets the baud rate setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xC5	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xC5	0x00	-	-

The host set the communication baud rate to 115200 by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xC4	0x04		

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xC4	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

HOST_BREAK_TIMEOUT Command (0xC2 / 0xC3) **<APP Layer>**

Purpose:
These commands are used to set/read the Break Timeout value used to exit from the Command Mode. The timeout or the CMD_EXIT_HOST_CONFIG_MODE stop the overload so the CONFIG/DEFAULT pin drives again the operating mode.

Design Tips:
In data mode, when a break is received on the UART, the radio modem host interface mode switches to configuration mode until the break timeout occurs. The timeout is reloaded either upon a command or a break reception. This configuration mode can be aborted thanks to the EXIT_HOST_CONFIG command.

Command SET: 0xC2; Length of the command: 0x02 bytes
Command GET: 0xC3; Length of the command: 0x01 byte

Parameter range:
0x00: The Command Mode is not enabled by the UART break reception. The CONFIG/DEFAULT pin always drives the mode.
0x01: A break reception will enable the Command Mode for maximum 1 * 100ms (it overloads the CONFIG/DEFAULT pin)
to
0xFE: A break reception will enable the Command Mode for maximum 254 * 100ms (it overloads the CONFIG/DEFAULT pin)
0xFF: A break reception will enable the command mode without any timeout (it overloads the CONFIG/DEFAULT pin).

Default parameter: 0x0A: A break reception will enable the Command Mode for a maximum of 10 * 100ms.

Example:
The host gets the Break Timeout setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xC3	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xC3	0x0A	-	-

The host set the Break Timeout to the infinite value by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xC2	0xFF		

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xC2	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

HOST_BYTE_TIMEOUT Command (0xC0 / 0xC1) **<APP Layer>**

Purpose:
These commands are used to set/read the minimum timeout between the end of two consecutive bytes. This variable is used by the Tx buffer to determine if a frame to send is complete or not. This timeout will give the authorization to send or not the frame on the next time slot. The unit is the "500µs"

Design Tips:
In data mode, when a byte time out occurs, the radio modem does not return an error code as the BYTE_TIMEOUT is not an error but an indication of the end of packet.

Command SET: 0xC0; Length of the command: 0x02 bytes
Command GET: 0xC1; Length of the command: 0x01 byte

Parameter range: 0x00: 125us
0x01: 500µs with steps of 500µs
to
0x23: Corresponds to the maximum (17.5ms)

Default parameter: 0x04. The default value is fixed to 4 x 500µs

Example:
The host gets the byte time-out setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xC1	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xC1	0x04	-	-

The host set byte time-out to 10ms (0x0A) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xC0	0x0A		

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xC0	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

HOST_DATA_END_CONDITION Command (0xDC / 0xDD) <APP Layer>

Purpose:
These commands are used to set/read the condition used for the end of data frame (only used in data mode).

Command SET: 0xDC; Length of the command: 0x02 bytes
Command GET: 0xDD; Length of the command: 0x01 byte

Parameter range:

- Bit 1: On_Channel:
Set to 1 to enable the end of data frame on the beginning of new free RF Channel.

WARNING:

This condition is true only if HOST_DATA_MODE_TX is set to "0".

- Bit 0: On_Byte_Timeout
If set to "1", the end of data frame is effective on a byte timeout.

Default parameter: 0x01: Byte timeout is used as end of data frame condition

Example:
The host gets the host data end condition setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xDD	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xDD	0x01		-

The host set end condition on Channel and on Byte timeout by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xDC	0x11		

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xDC	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

HOST_DATA_MODE Command (0xD0 / 0xD1) **<APP Layer>**

Purpose:
These commands are used to set/read the configuration of the data mode (cf. CMD_SEND_DATA and IND_RECEIVED_DATA)

Command SET: 0xD0 - [Data_Mode_Tx] - [Data_Mode_Rx]; Length of the command: 0x03 bytes

Command GET: 0xD1; Length of the command: 0x01 byte

Parameter range: Data_Mode_Tx: used to select which information must be sent from the host to the radio modem for the encapsulation.

- Bit 5: Check, redundancy check
- Bit 4: Retry, number of retry
- Bit 3: Tx Power, transmit output power
- Bit 2: Tx Destination, destination address
- Bit 1: Tx Group, used if destination address is multicast
- Bit 0: Tx Channel, channel ID

Data_Mode_Rx: used to select which information must be sent from the radio modem to the host to encapsulate the payload.

- Bit 7: Rx Destination; destination address
- Bit 6: Rx Group; destination group
- Bit 5: Rx Source; address of the transmitter
- Bit 4: Rx_Retry; retry value
- Bit 3: Channel ID where the RF reception has been made
- Bit 2: Rx_RSSI; RSSI output
- Bit 1: Frequency Error Indicator
- Bit 0: Rx_Size, length of the following payload (c.f. LAST_FRAME_INFO)

Default parameter: -

Example:
The host gets the host data mode setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xD1	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xD1	0x00	0x00	-

Please refer to CMD_SEND_DATA and IND_RECEIVED_DATA examples for the settings.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x03	0xD0	0x07	0x55	

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xD0	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

HOST_UART_SETTINGS Command (0xCC / 0xCD) **<APP Layer>**

Purpose:
These commands are used to set/read the UART configuration of the modem to insure compatibility with the largest set of hosts.

WARNING:

The host UART settings are changed after acknowledgement of the command.

Command SET: 0xCC; Length of the command: 0x02 bytes
Command GET: 0xCD; Length of the command: 0x01 byte

Parameter range:
Bit 4: [INV_RTS]
Value 0: RTS is active at "0"
Value 1: RTS is active at "1"
(Active means that the modem can receive more data or modem allows host to send more data)
Bit 3: [INV_CTS]
Value 0: CTS is active at "0"
Value 1: CTS is active at "1"
(Active means that the host can receive more data or host allows modem to send more data)
Bit 2: [PARITY_EN]
Value 0: Parity is disabled
Value 1: Parity is enabled
Bit 1: [PARITY_EVEN]
Value 0: Parity is odd
Value 1: Parity is even
Bit 0: [2_STOP_BITS]
Value 0: 1 stop bit
Value 1: 2 stop bits

Default parameter: 0x00

Example:
The host gets the host UART Settings by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xCD	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xCD	0x00		-

The host set the UART as following, active state of RTS to "1" and enables an odd parity with 2 stop bits by sending:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xCD	0x15	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xCC	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

LAST_BEACON_INFO Command (0x31) **<DLL Layer>**

Purpose:
This command is used to qualify the channel quality by getting the Receive Signal Strength indicator and the Frequency Error Indicator.

Design Tips:
The RSSI (Receive Signal Strength Indicator) measurement can be activated/ deactivated thanks to the command RSSI_MODE.

Command GET: 0x31; Length of the command: 0x02 bytes

Parameter range:
- Bit 5: Rx Source; address of the Beacon Server
- Bit 4: Not used
- Bit 3: Not used
- Bit 2: RSSI output if has been measured
- Bit 1: Frequency Error Indicator output if has been measured
- Bit 0: Beacon Status

Default parameter: -

Response Parameters:
- [Rx_SRC]: Beacon Server (Address 255 by default)
- [Rx_RSSI]: RSSI output (0 by default)
- [Rx_FEI_MSB]: FEI MSB output (0 by default)
- [Rx_FEI_LSB]: FEI LSB output (0 by default)
- [Sync_Status]: 0x00: Not Synchronized
  0x01: Synchronized
  0x02: Synchronized, but some beacon has been lost.

Example:
InfoMask = 0x26 = 0010 0110
Info= [RX_SRC] [RSSI][FEI_MSB] [FEI_LSB]

The host requests the information about the RSSI and the FEI by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x03	0x31	0x26	-	-

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From module to the host	0x05	0x31	[RX_ SRC]	[RSSI]	[FEI_ MSB]	[FEI_ LSB]

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

LAST_RX_FRAME_INFO Command (0x37) **<DLL Layer>**

Purpose:
This command is used to get information on the last received frame.

Design Tips:
The RSSI (Receive Signal Strength Indicator) measurement can be activated/ deactivated thanks to the command RSSI_MODE.

Command GET: 0x37; Length of the command: 0x02 bytes

Parameter range: InfoMask is used to select the desired information
- Bit 7: Destination Address
- Bit 6: Destination Group
- Bit 5: Source Address
- Bit 4: Number of retry
- Bit 3: Channel ID where the reception has been made
- Bit 2: RSSI output if has been measured
- Bit 1: Frequency Error Indicator output if has been measured
- Bit 0: Length of the last received payload

Default parameter: -

Response Parameters:
- [Rx_Dest]: Destination Address
- [Rx_Group]: Destination Group
- [Rx_Src]: Source Address
- [Rx_Retry]: Retry value
- [Rx_Channel]: Channel ID
- [Rx_RSSI]: RSSI output (0 by default)
- [Rx_FEI_MSB]: FEI MSB output (0 by default)
- [Rx_FEI_LSB]: FEI LSB output (0 by default)
- [Rx_Size]: Size of the payload

Example:
InfoMask = 0x55 = 0101 0101
Info= [Group][Retry][RSSI][Size]

The host requests the information about the group, the retry, the RSSI and the size of the last payload

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x37	0x55	-	-

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From module to the host	0x05	0x37	[Group]	[Retry]	[RSSI]	[Size]

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

LAST_TX_FRAME_INFO Command (0x33) **<DLL Layer>**

Purpose:
This command is used to get information on the last received frame.

Command GET: 0x33; Length of the command: 0x02 bytes

Parameter range: InfoMask is used to select the desired information
- Bit 7: Tx Status, transmission status
- Bit 6: Tx Size, Size of the transmitted frame
- Bit 5: Tx Check, redundancy check
- Bit 4: Tx Retry, value of the last try
- Bit 3: Tx Power, output power
- Bit 2: Tx Destination, destination address
- Bit 1: Tx Group, if the destination address was multicast
- Bit 0: Tx Channel, the last channel used for the transmission

Default parameter: -

Response Parameters:
- [TX_Status]:
  - 0x00: Frame acknowledged (transmission done)
  - 0x01: Frame not acknowledged (transmission done)
  - 0x02: Pending transmission (see TX_Retry)
  - 0x03: Not available
- [Tx_Size]: Size of the payload
- [Tx_Check]: Check mode
- [Tx_Retry]: Retry value
- [Tx_Power]: Output power value
- [Tx_Destination]: Destination address
- [Tx_Group]: group address
- [Tx_Channel]: Channel number of the last transmission

Example:
InfoMask = 0x65 = 1010 1001
Info= [Tx_Status][Tx_Check][Tx_Power][Tx_Channel]

The host requests the information about the Status, the check mode, the power and the Channel used for the last transmisson

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x33	0x65	-	-

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From module to the host	0x05	0x33	[Tx_ Status]	[Tx_ Check]	[Tx_ Power]	[Tx_ Channel]

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 t identify the error code.*

LIBIC_VERSION Command (0x11) **< APP Layer >**

Purpose:
This command can be used to retrieve the LibIC version.

Command GET: 0x11; Length of the command: 0x01 byte

Parameter range: -

Default parameter: Please contact Y-Lynx to obtain this information

Example:
The host gets the LibIC version by sending

	Byte 1	Byte 2			
From host to the module	0x01	0x11			

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x11	ASCII		

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the table 5 to identify the error code.*

LOAD_DEFAULT_PARAMETERS Command (0xF6) **<APP Layer>**

Purpose:
This command allows loading in the RAM memory a group of parameters stored in ROM memory. To load parameters from the Flash memory, the LOAD_PARAMETERS command will be used.

WARNING:

The parameters are loaded after acknowledgement of the command.

Design Tips:

LOAD_DEFAULT_PARAMETERS can also be replaced in UART mode, by sending 20 consecutives "break conditions".

Command LOAD: 0xF6; Length of the command: 0x02 bytes

Parameter range:
0x00: **All**
0x01: **User Data**
0x02: **Host Comm** (Baudrate, UART Settings, ByteTimeout, Break Timeout, Data Mode, CTS Threshold, GP1 Signal/Value, GP2 Signal/Value)
0x03: **Channel frequencies**
0x04: **Medium** (ChannelCount, ChannelDuration)
0x05: **RF Beacon** (Mode, Period, Listen Period, Max Beacon Lost, Max Beacon Track)
0x06: **RF Comm** (ACK Mode, Check Mode, Retry Count, Tx Retry restriction, User RF Frame Max Size)
0x07: **RF Config** (Power, Bitrate, Receiver Mode, RF Add, RF Destination Add, Tx Channel, NWK ID, RF Multicast, RF Destination Multicast, Application ID)
0x08: **Measure** (FEI Mode, RSSI Mode)

Default parameter: -

Example:
The host loads all groups by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xF6	0x00	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xF6	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

LOAD_PARAMETERS Command (0xF4) **<APP Layer>**

Purpose:
This command allows loading in the RAM memory a group of parameters stored in Flash memory. To load parameters from the ROM memory, the CONFIG/DEFAULT pin will be used at the power-up or the LOAD_DEFAULT_PARAMETERS command.

Design Tips:

LOAD_DEFAULT_PARAMETERS can also be replaced in UART mode, by sending 20 consecutives “break conditions”.

Command LOAD: 0xF4; Length of the command: 0x02 bytes

Parameter range:
0x00: **All**
0x01: **User Data**
0x02: **Host Comm** (Baudrate, UART Settings, ByteTimeout, Break Timeout, Data Mode, CTS Threshold, GP1 Signal/Value, GP2 Signal/Value)
0x03: **Channel frequencies**
0x04: **Medium** (ChannelCount, ChannelDuration)
0x05: **RF Beacon** (Mode, Period, Listen Period, Max Beacon Lost, Max Beacon Track)
0x06: **RF Comm** (ACK Mode, Check Mode, Retry Count, Tx Retry restriction, User RF Frame Max Size)
0x07: **RF Config** (Power, Bitrate, Receiver Mode, RF Add, RF Destination Add, Tx Channel, NWK ID, RF Multicast, RF Destination Multicast, Application ID)
0x08: **Measure** (FEI Mode, RSSI Mode)

Default parameter: -

Example:
The host loads all groups by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xF4	0x00	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xF4	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

MAX_BEACON_LOST Command (0x5A / 0x5B) **<DLL Layer>**

Purpose:
These commands are used to set/read the maximum beacon lost. When a beacon is not received, the client will wait until to receive a beacon frame or until the maximum beacon lost value is achieved. During this period of time, the client is still considered as synchronized.

Command SET: 0x5A; Length of the command: 0x02 bytes
Command GET: 0x5B; Length of the command: 0x01 byte

Parameter range: 0x00: Desynchronized after one beacon lost
0x0A: Desynchronized as soon as 11 consecutives beacons have been lost

Default parameter: 0x00: Desynchronized after one beacon lost

Example:
The host gets the maximum beacon lost setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x5B	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x5B	0x00	-	-

The host set the max beacon lost to “3”

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x5A	0x03	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x5A	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

MAX_BEACON_TRACK Command (0x5E / 0x5F) **<DLL Layer>**

Purpose:
These commands are used to set/read the maximum beacon track. When a client is considered as desynchronized, it switches in receiver mode until to receive a beacon frame or until the number of maximum beacon track is achieved. In this case the client switches in sleep mode. The MAX_BEACON_TRACK is based on the number of beacon period.

Command SET: 0x5E; Length of the command: 0x02 bytes
Command GET: 0x5F; Length of the command: 0x01 byte

Parameter range: 0x00: Switch automatically in sleep mode.
0x01: Set in receiver mode during 1 beacon period
0xFF: Set in receiver mode until a beacon is received, never goes to sleep mode

Default parameter: 0xFF: Set in receiver mode until a beacon is received, never goes to sleep mode

Example:
The host gets the Max beacon track setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x5F	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x5F	0xFF	-	-

The host set the max beacon track during “1” beacon period

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x5E	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x5E	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

NWK_ID Command (0x42 / 0x43) **<NWK Layer>**

Purpose:
These commands are used to set/read the Network identification of a device. The YLX-TRM module uses three address layers. To enable a communication between clients, the APP_ID and the NWK_ID must be identical. The Network ID is based on 6 bits.

Command SET: 0x42; Length of the command: 0x02 bytes
Command GET: 0x43; Length of the command: 0x01 byte

Parameter range: 0x00: Address 0
to
0x3F: Address 63

Default parameter: 0x3F, Network number 63

Example:
The host gets the network ID by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x43	-	-	-

The radio modem responds (the default ID)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x43	0x3F	-	-

The host set the network ID to 0x0D (13d) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x42	0x0D	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x42	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

POWER_MODE Command (0xD6 / 0xD7) **<APP Layer>**

Purpose:
These commands are used to set/read the power mode of the radio modem.

WARNING:

The power mode is changed after acknowledgement of the command.

Command SET: 0xD6; Length of the command: 0x02 bytes
Command GET: 0xD7; Length of the command: 0x01 byte

Parameter range:
0x00: The radio modem is set in normal mode PM_NORMAL
0x01: The radio modem is set in sleep mode: PM_SLEEP
- Only beacon are received
- Host Commands are decoded by the communication controller.
- RF data can be sent by the host.

0x02: The radio modem is set in shutdown mode: PM_SHUTDOWN
- RF transceiver is off
- Commands from the host are processed.

Default parameter: 0x00, PM_NORMAL

Example:
The host gets the radio modem power mode by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xD7	-	-	-

The radio modem responds (the default PM_NORMAL)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xD7	0x00	-	-

The host set the radio modem in shutdown mode by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xD6	0x02	-	-

To acknowledge the command the modem sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xD6	-	-	-

If an error occurred during the communication, the modem response the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RECEIVED_DATA Indicator (0xD3) **<APP Layer>**

Purpose:
This indicator is used to receive data from a remote node when the configuration mode is enabled (cf. CONFIG/DEFAULT pin).

The RECEIVED_DATA indicator depends on the Data_Mode_Rx settings.

Design Tips:

In configuration mode and during the radio modem initialization, a way to avoid the reception of RF data frames (RECEIVED_DATA) is to switch the modem in sleep mode (PM_SLEEP) and then back in normal mode (PM_NORMAL) after the end of the initialization phase. This can be done thanks to the POWER_MODE command.

Command: 0xD3; Length of the command: undefined

Parameter range: Data_Mode_Rx: used to select which information must be sent from the radio modem to the host to encapsulate the payload.

- Bit 7: Rx Destination; destination address
- Bit 6: Rx Group; destination group
- Bit 5: Rx Source; address of the transmitter
- Bit 4: Rx_Retry; retry value
- Bit 3: Channel ID where the RF reception has been made
- Bit 2: Rx_RSSI; RSSI output
- Bit 1: Frequency Error Indicator
- Bit 0: Rx_Size, length of the following payload (c.f. LAST_FRAME_INFO)

Example: Settings of Data_Mode_Rx= 0x55 = 0101 0101

- Rx_Group: enabled
- Rx_Retry: enabled
- Rx_RSSI: enabled
- Rx_Size: enabled

The received payload is 0xAA with the destination group 0x03, number of retry equal to 0x01, RSSI of 0x03 and payload's size of 0x01.

The host received data (0xAA) according to Data_Mode_Rx settings

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
From module to the host	0x06	0xD3	0x03	0x01	0x03	0x01	DATA (0xAA)

RECEIVED_RESPONSE Indicator (0xD9) **<APP Layer>**

Purpose:
This indicator is used to receive the response from a remote node when a remote command has been sent (SEND_COMMAND).

The SEND_COMMAND command and then the response (RECEIVED_RESPONSE) can be used only if the remote wireless module (node) has its remote command mode enabled (REMOTE_CMD_MODE).

WARNING:

This command is only available with the firmware version 1.03 or later.

Command: 0xD9; Length of the command: undefined

Parameter range: Length of the remote command + remote command + Response

With remote command:
0x10: GET_VERSION
0x11: GET_LIBIC_VERSION
0xE2/0xE3: SET/GET_GP1_VALUE
0xE6/0xE7: SET/GET_GP2_VALUE
0x76/0x77: SET/GET_REMOTE_CMD_MODE
0x78/0x79: SET/GET_ALL_CHANNELS_TYPE
0x3C/0x3D: SET/GET_CHANNEL_TYPE

Example:

Read the value of the remote GP1, (the remote GP1 has been configured as input).

The host sends the command (SEND COMMAND; 0xE3) to get the remote GP1 value, the length of this command is 1 byte (0x01)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From host to the module	0x03	0xD8	0x01	0xE3	-	-

To module sends back the response received (RECEIVED_RESPONSE; 0xD9) from the remote module, the value of GP1 is “1” (high level)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x04	0xD9	0x02	0xE3	0x01

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	0x07	-	-

REMOTE_CMD_MODE Command (0x76 / 0x77) **<APP Layer>**

Purpose:
These commands are used to set/read the remote command mode. To accept remote command from another node, the module has first to be enabled remote command mode option.

WARNING:

This command is only available with the firmware version 1.03 or later.

Command SET: 0x76; Length of the command: 0x02 bytes
Command GET: 0x77; Length of the command: 0x01 byte

Parameter range: 0x00: Remote command mode disabled, only SEND_ECHO_DATA and SET/GET_REMOTE_CMD_MODE are enabled.
0x01: Remote command mode enabled

Default parameter: 0x00: Remote command mode disabled

Example:
The host gets the remote command mode setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x77	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x77	0x00	-	-

The host enables the remote command mode (to the connected module) by

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x76	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x76	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RESET_MODE Command (0xDE / 0xDF) **<APP Layer>**

Purpose:

These commands are used to define the source of the loaded configuration (Y-Lynx / User) and on which condition (conditional / unconditional) after a reset.

Command SET: 0xDE; Length of the command: 2 bytes
Command GET: 0xDF; Length of the command: 1 byte

Parameter range: 0x00: Load Flash parameters if Config/default pin is set to "1" unless load ROM parameters (conditional execution)
0x01: Load Flash parameters if Config/default pin is set to "0", unless load ROM parameters (conditional execution)
0x02: Load Flash parameters (unconditional execution)
0x03: Load ROM parameters (unconditional execution)

Default parameter: 0x00, Load Flash parameters if Config/default pin is set to "1"

Example:

The host gets the reset mode setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xDF	-	-	-

The radio modem responds (default reset mode)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xDF	0x00	-	-

The host set the reset mode to load Flash parameters (user parameters) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xDE	0x02	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xDE	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_ACK_MODE Command (0x50 / 0x51) **<DLK Layer>**

Purpose:
These commands are used to set/read the RF Addressed Mode. The Y-Lynx radio modem offers two types of addressing mode: with or without acknowledge. Acknowledgements are done in the same channel and allow retry mechanism to take place in the following channels.

Design Tips:

As the acknowledgments are done in the same channel, there is less time for the RF data transmission.

Command SET: 0x50; Length of the command: 2 bytes
Command GET: 0x51; Length of the command: 1 byte

Parameter range: 0x00: Acknowledge Mode disabled
0x01: Acknowledge Mode enabled

Default parameter: 0x01, Acknowledge Mode enabled

Example:
The host gets the Acknowledge Mode setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x51	-	-	-

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x51	0x01	-	-

The host disables the Acknowledge Mode (0x00) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x50	0x00	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x50	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_ADDRESS Command (0x44 / 0x45) **<DLK Layer>**

Purpose:
These commands are used to set/read the Device Address. The YLX-TRM module uses three address layers. To enable a communication between clients, the APP_ID and the NWK_ID must be identical. The RF_ADDRESS must be unique in the network and application topology, so the same address can exist if the network or the application identification is different.

Command SET: 0x44; Length of the command: 2 bytes
Command GET: 0x45; Length of the command: 1 byte

Parameter range: 0x00: Address 0
to
0xFE: Address 254

Default parameter: 0x7F, Device address 127

Example:
The host gets the RF address by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x45	-	-	-

The radio modem responds

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x45	0x00	-	-

The host set the device address to 0xDD (221d) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x44	0xDD	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x44	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_BITRATE Command (0x62 / 0x63) **<PHY Layer>**

Purpose:
These commands are used to set/read the RF data rate (data rate transmitted over the air) of the module. When modified, the RF filter bandwidth and the RF frequency deviation are automatically adjusted.

Command SET: 0x62; Length of the command: 0x02 bytes
Command GET: 0x63; Length of the command: 0x01 byte

Parameter range:
0x00: 4.8 kbit/s
0x01: 9.6 kbit/s
0x02: 19.2 kbit/s
0x03: 38.05 kbit/s
0x04: 76.1 kbit/s
0x05: 152.34 kbit/s

Default parameter: 0x04: Bit rate of 76.1 kbit/s

Example:
The host gets the RF bit rate setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x63	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x63	0x04	-	-

The host set the bit rate to 9.6kbit/s by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x62	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x62	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_CHECK_MODE Command (0x52 / 0x53) **<DLK Layer>**

Purpose:
These commands are used to set/read the configuration of the RF redundancy check. The Y-Lynx radio modem can be configured with four different redundancy check modes: None, LRC, CRC8 or CRC16.

Command SET: 0x52; Length of the command: 0x02 bytes
Command GET: 0x53; Length of the command: 0x01 byte

Parameter range: 0x00: None
0x01: LRC
0x02: CRC8
0x03: CRC16

Default parameter: 0x02, CRC8

Example:
The host gets the "RF Check Mode" setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x53	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x53	0x00	-	-

The host set the flow control with a CRC8 (0x02) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x52	0x02	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x52	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_DEST_ADDRESS Command (0x4A / 0x4B) **<DLK Layer>**

Purpose:
These commands are used to set/read the destination address of a device. The YLX-TRM module uses three address layers –Application identification (APP_ID), Network Identification (NWK_ID), and Device Identification (DEV_ID). The RF_DEST_ADDRESS command assigns a destination address to the Device Identification and enables it to communicate only with one module having the same APP_ID and NWK_ID.

Command SET: 0x4A; Length of the command: 0x02 bytes
Command GET: 0x4B; Length of the command: 0x01 byte

Parameter range: Unicast destination address
0x00: Address 0
to
0xFE: Address 254
Multicast destination address (c.f. RF_MULTICAST)
0xFF: Address 255

Default parameter: 0x7F, Address 127

Example:
The host gets the destination address setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x4B	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x4B	0x00	-	-

The host set the destination address to 0xBD to send a message to the device #189

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x4A	0xBD	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x4A	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_DEST_MULTICAST Command (0x4E / 0x4F) **<NWK Layer>**

Purpose:
These commands are used to set/read the multicast address used by the node to transmit a message to one or more group. The Communication Controller offers 8 different groups. The node is registered to a group if the group address bit is enabled (enable =1).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Group 7	Group 6	Group 5	Group 4	Group 3	Group 2	Group 1	Group 0

To transmit a message to one or more group, the RF_DEST_ADDRESS must be set to MULTICAST (0xFF), please refer to RF_DEST_ADDRESS.

Retries and acknowledgements do not take place when multicast communication is used.

Command SET: 0x4E; Length of the command: 0x02 bytes
Command GET: 0x4F; Length of the command: 0x01 byte

Parameter range:
0000 0001b: The node sends the message to the group 0 only
0000 0010b: The node sends the message to the group 1 only
0000 0100b: The node sends the message to the group 2 only
……. …….
1111 1111b: The node sent the message to all the groups

Default parameter: 0x00

Example:
The host gets the actual setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x4F	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x4F	0xFF	-	-

The host sends a message to the groups 1,2,5 and 7(1010 0110)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x4E	0xA6	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x4E	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_FRAME_MAX_SIZE Command (0xCB) **<DLK Layer>**

Purpose:

This command is used to get the absolute maximum size of a RF frame. The maximum RF frame size corresponds to the minimum value between:

- The value of USER_RF_FRAME_MAX_SIZE

and

- The maximum size calculated by the radio modem. This calculation depends on the actual RF parameters (RF_BITRATE, CHANNEL_DURATION, ACK_MODE, CHECK_MODE, …).

Command GET: 0xCB; Length of the command: 0x01 byte

Parameter range: -

Default parameter: -

Example:

The host gets the maximum size of the RF frame by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xCB	-	-	-

The radio modem responds 128 bytes (0x80)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xCB	0x80	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_MULTICAST Command (0x4C / 0x4D) **<NWK Layer>**

Purpose:
These commands are used to set/read the multicast address. The Communication Controller offers 8 different groups; a node can be register to any of these groups. The node is registered to a group if the group address is enabled (enable =1). The multicast address is an additional address to the node; its unicast address is still valid (RF_ADDRESS).

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Group 7	Group 6	Group 5	Group 4	Group 3	Group 2	Group 1	Group 0

Command SET: 0x4C; Length of the command: 0x02 bytes
Command GET: 0x4D; Length of the command: 0x01 byte

Parameter range:
0000 0000b: Unicast only
0000 0001b: The node is registered to the group 0 only
0000 0010b: The node is registered to the group 1 only
0000 0100b: The node is registered to the group 2 only
0000 0110b: The node is registered to the group 1 and 2
0000 1000b: The node is registered to the group 3
1111 1110b: The node is registered to all the groups.

Default parameter: 0x00, No group registered (unicast only)

Example:
The host gets the actual setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x4D	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x4D	0x00	-	-

The host registers the node to the groups 1,2,5 and 7(1010 0110)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x4C	0xA6	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x4C	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_POWER Command (0x60 / 0x61) **<PHY Layer>**

Purpose:
These commands are used to set/read the RF output power.

Command SET: 0x60; Length of the command: 0x02 bytes
Command GET: 0x61; Length of the command: 0x01 byte

Parameter range: 0x00: Power level 0 (min)
0x01: Power level 1
0x02: Power level 2
0x03: Power level 3 (max)

Default parameter: 0x03: Power level 3 (maximum)

Example:
The host gets the RF power setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x61	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x61	0x03	-	-

The host set the power level to the minimum by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x60	0x00	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x60	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_RECEIVER_MODE Command (0x6A / 0x6B) **<PHY Layer>**

Purpose:
These commands are used to set/read the receiver sensitivity. The choice depends on the environment, if it is noisy; the high linearity mode should be selected.

Command SET: 0x6A; Length of the command: 0x02 bytes
Command GET: 0x6B; Length of the command: 0x01 byte

Parameter range: 0x00: High Sensitivity Mode
0x01: High Linearity Mode

Default parameter: 0x00: High Sensitivity Mode

Example:
The host gets the Receiver Mode setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x6B	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x6B	0x00	-	-

The host set the High Linearity Mode by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x6A	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x6A	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RF_RETRY_COUNT Command (0x5C / 0x5D) **<DLK Layer>**

Purpose:
These commands are used to set/read the number of retries that can be sent for a given unicast RF packet when acknowledge mode is enabled. After transmitting a packet, the transmitting module expects to receive an acknowledgement from the receiving module. If the acknowledgement is not received, the original packet is transmitted again. The RF packet is transmitted repeatedly until an acknowledgement is received or until the packet is sent RF_RETRY_COUNT+1 times.

The value of the retry is sent by RF to a remote node, the receiver can obtain the value via the RX_RETRY field in the LAST_FRAME_INFO.

GP2 can be configured to obtain acknowledge information (GP2_TX_ACK)

Command SET: 0x5C; Length of the command: 0x02 bytes
Command GET: 0x5D; Length of the command: 0x01 byte

Parameter range: 0x00: No retry
0x07: Retry up to 7 times

Default parameter: 0x00: No retry

Example:
The host gets the retry count by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x5D	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x5D	0x00	-	-

The host set the 3 retries by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x5C	0x03	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x5C	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

RSSI_MODE Command (0x2A / 0x2B) **<DLK Layer>**

Purpose:
These commands are used to enable/disable the Receive Signal Strength Indicator (RSSI). Four modes are available, no measure, measure on received beacon signal, measure on every received data packets (from server or others clients) and measure on all received packets (beacon and data).

The RSSI provides 2 bits information. The description is described in the table below:

Diagnostic	RSSI output	Description
Very Poor	00	RF input < -110dBm
Poor	01	-110dBm < RF input <-105dBm
Good	10	-105dBm < RF input < -100dBm
Excellent	11	-100dBm < RF input

Command SET: 0x2A; Length of the command: 0x02 bytes
Command GET: 0x2B; Length of the command: 0x01 byte

Parameter range:
0x00: No RSSI measurement
0x01: RSSI measurement on received beacon only
0x02: RSSI measurement on every received data packets
0x03: RSSI measurement on all received packets

Default parameter: 0x03: RSSI measurement on all received packets

Example:
The host gets the RSSI Mode setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0x2B	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x2B	0x00	-	-

The host set the RSSI measurement on received beacon signal by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0x2A	0x01	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0x2A	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

SAVE_PARAMETERS Command (0xF2) **<APP Layer>**

Purpose:
This command allows saving in the Flash memory a group of parameters, stored in RAM.

WARNING:

This command's execution takes some time due to the flash access and needs to be completed in highest priority. This may conduce to a beacon lost.

Command: 0xF2; Length of the command: 0x02 bytes

Parameter range:
0x00: **All**
0x01: **User Data**
0x02: **Host Comm** (Baudrate, UART Settings, ByteTimeout, Break Timeout, Data Mode, CTS Threshold, GP1 Signal/Value, GP2 Signal/Value)
0x03: **Channel frequencies**
0x04: **Medium** (ChannelCount, ChannelDuration)
0x05: **RF Beacon** (Mode, Period, Listen Period, Max Beacon Lost, Max Beacon Track)
0x06: **RF Comm** (ACK Mode, Check Mode, Retry Count, Tx Retry restriction, User RF Frame Max Size)
0x07: **RF Config** (Power, Bitrate, Receiver Mode, RF Add, RF Destination Add, Tx Channel, NWK ID, RF Multicast, RF Destination Multicast, Application ID)
0x08: **Measure** (FEI Mode, RSSI Mode)

Default parameter: -

Example:
The host saves all groups by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xF2	0x00	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xF2	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

SEND_COMMAND Command (0xD8) **<APP Layer>**

Purpose:
This command is used to send command to a remote node when the configuration mode is enabled (cf. CONFIG/DEFAULT pin).

The SEND_COMMAND command can be used only if the remote wireless module (node) has its remote command mode enabled (REMOTE_CMD_MODE).

WARNING:

This command is only available with the firmware version 1.03 or later.

Command: 0xD8; Length of the command: undefined

Parameter range:
0x10: GET_VERSION
0x11: GET_LIBIC_VERSION
0xE2/0xE3: SET/GET_GP1_VALUE
0xE6/0xE7: SET/GET_GP2_VALUE
0x76/0x77: SET/GET_REMOTE_CMD_MODE
0x78/0x79: SET/GET_ALL_CHANNELS_TYPE
0x3C/0x3D: SET/GET_CHANNEL_TYPE

Other command receive an undefined command error (0x02 0x03 0x07)

Example:

Read the value of the remote GP1, (the remote GP1 has been configured as input).

The host sends the command (0xE3) to get the remote GP1 value, the length of this command is 1 byte (0x01)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From host to the module	0x03	0xD8	0x01	0xE3	-	-

To module sends back the response received (0xD9) from the remote module, the value of GP1 is “1” (high level)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x04	0xD9	0x02	0xE3	0x01

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	0x07	-	-

SEND_DATA Command (0xD2) **<APP Layer>**

Purpose:
This command is used to send data to a remote node when the configuration mode is enabled (cf. CONFIG/DEFAULT pin).

The SEND_DATA command depends on the Data_Mode_Tx settings.

Design Tips:

In case of an ERR_OUTOFRANGE_ARGUMENT (0x03 0x03 0x04 0xxx), the argument 0xxx indicates the erroneous argument in the Data_Mode_Tx bit field.

Command: 0xD2; Length of the command: undefined

Parameter range: Data_Mode_Tx: used to select which information must be sent from the host to the radio modem for the encapsulation.

- Bit 5: Check, redundancy check
- Bit 4: Retry, number of retry
- Bit 3: Tx Power, transmit output power
- Bit 2: Tx Destination, destination address
- Bit 1: Tx Group, used if destination address is multicast
- Bit 0: Tx Channel, channel ID

Example:

Settings of Data_Mode_Tx= 0x07 = 0000 0111

- Tx_dest : enabled
- Tx_group: enabled
- Tx_channel: enabled

Data to send 0xAA to the destination 0xFF (multicast address), to group0 and group1, so 0x03 and on the channel 0x05

The host sends data (0xAA) according to Data_Mode_Tx settings

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From host to the module	0x05	0xD2	0xFF	0x03	0x05	DATA (0xAA)

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xD2	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

SEND_ECHO_DATA Command (0xD4) **<APP Layer>**

Purpose:
This command is used to send an RF data frame to a hostless modem which will echo it back.

Command: 0xD4 Length of the command: undefined

Parameter range: Data_Mode_Tx: used to select which information must be sent from the host to the radio modem for the encapsulation.
- Bit 5: Check, redundancy check
- Bit 4: Retry, number of retry
- Bit 3: Tx Power, transmit output power
- Bit 2: Tx Destination, destination address
- Bit 1: Tx Group, used if destination address is multicast
- Bit 0: Tx Channel, channel ID

Example:

Settings of Data_Mode_Tx= 0x05 = 0000 0101, Please refer to the SET_HOST_DATA_MODE command
- Tx_dest : enabled
- Tx_group: disabled
- Tx_channel: enabled

Data to send 0xAA, 0xBB to the destination 0x09 (address 9), and on the channel 0x08

The host sends data (0xAA, 0xBB) according to current Data_Mode_Tx settings

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From host to the module	0x05	0xD4	0x09	0x08	DATA (0xAA)	DATA (0xBB)

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xD4	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

STATUS Command (0x13) **<APP Layer >**

Purpose:
This command asks a status to the module, the answer is the power-on self test status, the self test is not executed in response at this command.

Command: 0x13 Length of the command: 0x01 byte

Parameter range: -

Default parameter: -

Response parameters: 0x00: Status Ok
Others please contact Y-Lynx with the status value.

Example:
The host requests the status of the radio modem by sending

	Byte 1	Byte 2				
From host to the module	0x01	0x13				

The radio modem responds (the last self test status)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x13	0x00	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

TX_CHANNEL Command (0xC6 / 0xC7) **<DLK Layer>**

Purpose:
These commands are used to set/read the channel restriction for the transmission. Thanks to the SET command, the RF transmission is limited to only one specific channel. That can be the channel "0" (Beacon channel) if the beacon period is not equal to 1 (the beacon is sent at the beginning of each sequence, so the channel "0" is not available for RF data transmission).

Command SET: 0xC6; Length of the command: 0x02 bytes
Command GET: 0xC7; Length of the command: 0x01 byte

Parameter range:
0x00: RF transmission on channel 0 (if beacon period >1)
0x01: RF transmission on channel 1
……..
0xxx: RF transmission is limited to the maximum defined channel (if CHANNEL_COUNT is different from MAX_CHANNEL_COUNT 0x32)

0x31: RF transmission is limited to channel 49 (MAX_CHANNEL_COUNT – 1)

0xFF: No channel restriction

Default parameter: 0xFF: No channel restriction

Example:
The host gets the actual setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xC7	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xC7	0x00	-	-

The host set the restriction of the transmission to channel 3 only by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xC6	0x03	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xC6	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

TX_RETRY_RESTRICTION Command (0xC8 / 0xC9) **<DLK Layer>**

Purpose:
These commands are used to set/read channel restriction for the retries. To use this command, the acknowledge mode needs to be enabled and the retry count needs to be set to a value different of 0.

Command SET: 0xC8; Length of the command: 0x02 bytes
Command GET: 0xC9; Length of the command: 0x01 byte

Parameter range: 0x00: Channel restriction for the retries. If a channel restriction has been set (TX_CHANNEL), the retry will be occurring on the same channel on the next sequence. If the first attempt has occurred on channel "3", the retry will occur on channel "3" of the next sequence.

0x01: No channel restriction for the retries. The retry will be occurring on the next channel. If the first attempt has occurred on channel "3", the retry will occur on channel "4".

Default parameter: 0x01, No channel restriction

Example:
The host gets the actual setting by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xC9	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xC9	0x00	-	-

The host set no restriction on retries by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xC8	0x01	-	-

To acknowledge the command the modem sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xC8	-	-	-

If an error occurred during the communication, the modem response the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

USER_DATA Command (0xF0 / 0xF1) **<APP Layer>**

Purpose:
These commands are used to set/read 4 bytes in a RAM memory as the user version number or any other useful information. To save it in Flash use the SAVE_PARAMETERS command

Command SET: 0xF0; Length of the command: 0x05 bytes
Command GET: 0xF1; Length of the command: 0x01 bytes

Parameter range: Byte0: User defined
Byte1: User defined
Byte2: User defined
Byte3: User defined

Default parameter: 0x00

Example:
The host read the four byte values by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Bate 5	Byte 6
From host to the module	0x01	0xF1	-	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From module to the host	0x05	0xF1	0x00	0x00	0x00	0x00

The host set a version number (V1.0) by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
From host to the module	0x05	0xF0	0x56	0x01	0x2E	0x00

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xF0	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

USER_RF_FRAME_MAX_SIZE Command (0xCE / 0xCF) **<DLK Layer>**

Purpose:
These commands allow the modem to consider its buffer of reception as full as soon as this size is reached. So, the CTS signal can be raised. When the host sends a frame with a size lower than the current USER_RF_FRAME_MAX_SIZE, one of the others conditions will be used to determine the end of frame:

- The BYTE_TIMEOUT if enabled
- A new RF channel start if enabled
- The CONFIG/DEFAULT pin change from Data to Config
- The RF_FRAME_MAX_SIZE calculated by the Communication Controller

Command SET: 0xCE; Length of the command: 0x02 bytes
Command GET: 0xCF; Length of the command: 0x01 byte

Parameter range: 0x00: 0 bytes
to
0x80: 128 bytes

Default parameter: 0x80

Example:
The host gets the maximum size of the frame by sending

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x01	0xCF	-	-	-

The radio modem responds (the default setting)

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0xCF	0x80		-

The host set the maximum size of the frame to 50 (50 bytes) by sending:

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From host to the module	0x02	0xCE	0x32	-	-

To acknowledge the command the module sends back the following frame

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x01	0xCE	-	-	-

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

VERSION Command (0x10) **<APP Layer>**

Purpose:
This command is used to obtain the module version number; this number has been fixed by Y-Lynx: (The response is given in ASCII)

Command GET: 0x10; Length of the command: 0x01 byte

Parameter range: -

Default parameter: Please contact Y-Lynx to obtain this information

Example:
The host sends a version request by sending

	Byte 1	Byte 2			
From host to the module	0x01	0x10			

The radio modem responds

	Byte 1	Byte 2	Byte 3		
From module to the host	0x02	0x10	ASCII		

If an error occurred during the communication, the module responses the following frame.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
From module to the host	0x02	0x03	Err**	-	-

***Err: Please refer to the tables 3.5 and 3.6 to identify the error code.*

3 APPENDIX

Documentation History

Revision	Modifications	Date
Rev 1.2	New release of the wireless communication protocol	Jan 2011
Rev 1.1	Minor bugs correction	Oct 2010
Rev 1.0	Original version	May, 2008

Related Products and Documents

Related Documentations

Description	Type of document
YLX-TRM8053-025-05	Data Sheet
YLX-TRM8053-500-05	Data Sheet
Errata Sheet	See www.y-lynx.com for details

YLX-TRM8053-025-05 Radio Modem

Description	Ordering Number
YLX-TRM8053-025-05 Drop'in Modem	YLX-TRM8053-025-05DIT

YLX-TRM8053-500-05 Radio Modem

Description	Ordering Number
YLX-TRM8053-500-05 Drop'in Modem	YLX-TRM8053-500-05DIT

Development Kit

Description	Ordering Number
Radio Modem Starter Kit for YLX-TRM8053-025-05	RMSK-TRM8053-025-05
Radio Modem Starter Kit for YLX-TRM8053-500-05	RMSK-TRM8053-500-05

Contact Information

Address:

Y-Lynx Sarl

Rue de Galilée15
CH - 1400 YVERDON
SWITZERLAND

Phone: +41 24 423 92 05
Email: info@y-lynx.com
Web site: http://www.y-lynx.com

