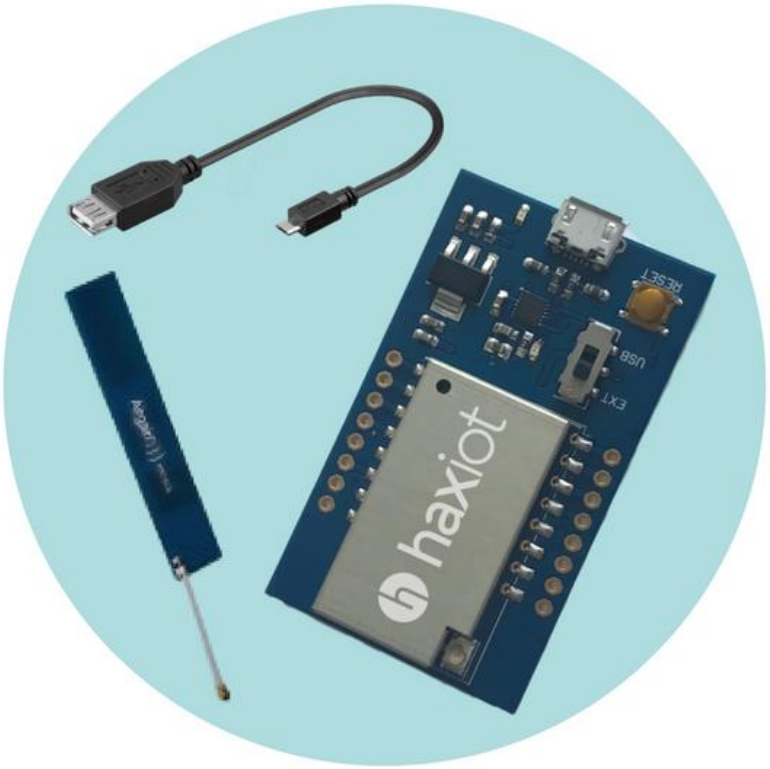
HXC Client User Guide



Version 2.01

Disclaimer and Copyright Notice

LoRa is a registered trademark of Semtech Corporation
LoRaWAN is a registered trademark of the Lora Alliance
Haxiot is a registered trading name of Iotek Systems, LLC

Table of Contents

List of Figures

List of Tables

1 Introduction

The Haxiot HXC Client module supports the LoRa radio technology and the LoRaWAN protocol stack. This enables long-range, low-power devices to be remotely administered. The technology allows devices to connect with gateways up to 15km. The module requires a gateway and a Network Server.

The HXC client module provides easy integration to devices with the HXC Embedded API. Service Providers, Enterprises, and application providers can use the HXC Zero Touch API to activate customer devices on Haxiot cloud services.

There are two variations of the HXC Client Module, 400MHz (Model: HXC400) and 900MHz (Model: HXC900). Both variants share a common footprint and software API for global coverage in a range of LoRaWAN channel bands.

Module features

- LoRaWAN and LoRa modes
- UART interface
- AES128- bit encryption
- Class A and Class C
- Industry standard AT command set
- Embedded API for device integration
- OTAA and ABP authentication
- Public or private networks
- Standard or custom regional channel plans
- Global roaming ready LoRaWAN address
- Zero-Touch Provisioning API for factory activation
- Vibration resistant to industrial devices
- Wide range of supply voltages 2.2 - 3.6V
- Ultra-low power sleep mode (11uA)
- FCC (USA) and IC (Canada) certified (HXC900)

2 Physical Characteristics

Table 2-1: Absolute Maximum Ratings

Description	Symbol	Min	Typ.	Max	Units
Supply voltage	Vcc	0.5	3.3	3.9	V
Control voltages	V_{TX}, V_{RX}	-0.2	-	-	V
RF Input Power	RFI_{pwr}	-	-	+10	dBm
RF output power	RFO_{pwr}	+13	-	+20	dBm
Temperature	T	-40	-	+85	°C
Shock	-	3x fall of 100cm			

Table 2-2: Operating Range

Description	Symbol	Min	Max	Units
Supply voltage	Vcc	2.5	3.6	V
Control voltages	V_{TX}, V_{RX}	-0.2	-	V
RF Input Power	RFI_{pwr}	-	+10	dBm
RF Output Power	RFO_{pwr}	+14	+20	dBm
Temperature	T	-40	+85	°C

Table 2-3: Power Consumption

Description	Symbol	Min	Typ	Max	Units
Transmit	Tx	-	126	-	mA
Receive	Rx	-	17	-	mA
Idle	Idle	-	4	-	mA
Sleep	Sleep	10	11	12.5	uA

Power consumption is based on ambient temperature of 25°C and 3.3V supply voltage.

Table 2-4: Physical Properties

Description	Symbol	Min	Typ	Max	Units
Length	L	-	30.5	-	mm
Width	W	-	15.0	-	mm
Height	H	-	4	-	mm
Weight	Wg	-	0.16	-	oz

3 Radio Performance

The HXC Client utilizes a +20dB LoRa radio chipset from Semtech for transmitting and receiving. The external radio connection for the antenna uses a Hirose U.FL micro coaxial cable connector with 50ohm impedance matching. The radio has been optimized for transmission and reception across the 902-928Mhz ISM frequency band.

3.1 Antenna Selection

The HXC client modules require external, dipole antennas. Haxiot recommends Airgain flexible F910D 863-930Mhz antenna with a peak gain of 2.2dB.

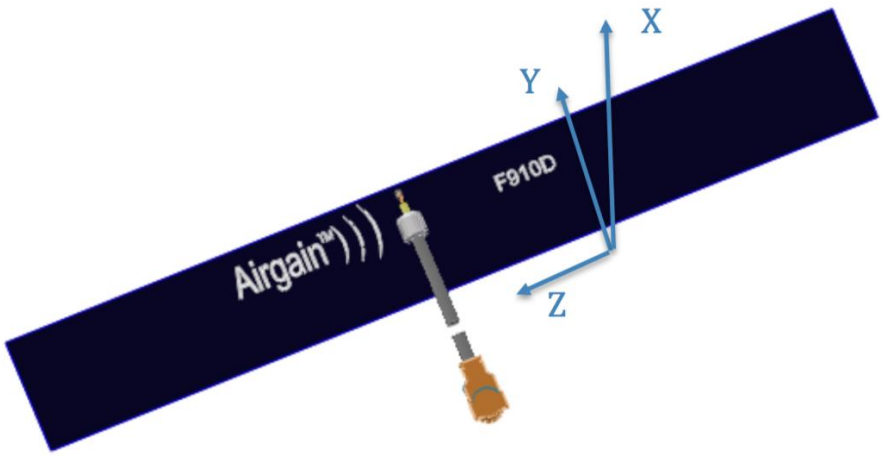


Figure 3-1: Airgain F910D measurement axes

Figure 3-2: F910D radiation patterns at 915Mhz

4 Operating Modes

This section provides an overview of the operating modes and regional parameters for the HXC series module.

4.1 LoRaWAN Class A

The HXC series module supports Class A and Class C modes of operation. In Class A operating mode, the HXC module can transmit anytime but will only receive during the Receive Window 1 and Receive Window 2. When not sending, receiving or processing AT commands, the module will be in low-power sleep mode. Executing an AT command will wake up the module.

4.2 LoRaWAN Class C

In Class C mode, the HXC module is in Rx mode if not sending data in Tx mode. The device will receive packets on Receive Window 1 only after transmission of a packet at the RX1 Delay Window timeslot. At all other times, the module will be in Rx mode using the Receive Window 2 frequency and data rate.

4.3 LoRa Modem

The modem can be in LoRaWAN mode (default) or LoRa mode. In the LoRa mode, the HXC module supports single packet sending or continuous receiving of LoRa packets without the LoRaWAN protocol. This mode allows for sending/receiving of 3rd party protocols over LoRa modulation that do not use the LoRaWAN standard.

4.4 Test Mode

This mode supports a continuous-transmit command for testing purposes only. When the Continuous Transmit command is executed the HXC module will continuously transmit an un-modulated RF signal. This mode is ideal for testing and tuning antennas for devices being fitted for enclosures.

4.5 LoRaWAN Regional Parameters

4.5.1 HXC400 Parameters

The tables below show all the necessary parameters for HXC Client Module 400MHz (Model: HXC400).

Table 4-1: HXC400 Default Parameters

Parameters	Default	Modifiable
Device EUI	Globally Unique EUI-64	Yes
Device Address	0	Yes
Application EUI	0	Yes
Application Key	0	Yes
Application Session Key	0	Yes
Network Session Key	0	Yes
Maximum Datarate	DR5	No
Minimum Datarate	DR0	No
Default Datarate	DR5	Yes
Maximum Transmit Power	TX0	No
Minimum Transmit Power	TX5	No
Default Transmit Power	TX0	Yes
Maximum Receive Window	3 seconds	No
Receive Window 1 Delay	1 second	Yes
Receive Window 2 Delay	2 seconds	Yes
Join Rx Window 1 Delay	5 seconds	Yes
Join Rx Window 2 Delay	6 seconds	Yes
Rx Window 2 Frequency	471.3MHz	Yes
Rx Window 2 Datarate	DR3	Yes
Default Channel Configurations	<CH No>, <Freq(MHz)>, <DRmax>, <DRmin> 0, 471.5, 5, 0 1, 471.7, 5, 0 2, 471.9, 5, 0 3, 472.1, 5, 0 4, 472.3, 5, 0 5, 472.5, 5, 0 6, 472.7, 5, 0 7, 472.9, 5, 0	Yes
Maximum Possible Number of Channels	16	No

Table 4-2: HXC400 Data Rate Table

Datarate	Spreading Factor	Payload Size (Bytes)	Bandwidth	Bit/s
DR0	SF12	51	125KHz	250
DR1	SF11	51		440
DR2	SF10	51		980
DR3	SF9	115		1760
DR4	SF8	222		3125
DR5	SF7	222		5470

Table 4-3: HXC400 Power Table

Tx Power	Configuration (dB)
TX0	20
TX1	14
TX2	11
TX3	8
TX4	5
TX5	2

4.5.2 HXC900 Parameters

The tables below show all the necessary parameters for HXC900.

Table 4-4: HXC900 Default Parameters

Parameters	Default	Modifiable
Device EUI	Globally Unique EUI-64	Yes
Device Address	0	Yes
Application EUI	0	Yes
Application Key	0	Yes
Application Session Key	0	Yes
Network Session Key	0	Yes
Tx Minimum Datarate	DR0	No
Tx Maximum Datarate	DR4	No
Default Tx Datarate	DR0	Yes
Rx Minimum Datarate	DR8	No
Rx Maximum Datarate	DR13	No
Maximum Transmit Power	TX5	No
Minimum Transmit Power	TX10	No
Default Transmit Power	TX5	Yes
Maximum Receive Window	3 seconds	No
Receive Window 1 Delay	1 second	Yes
Receive Window 2 Delay	2 seconds	Yes
Join Rx Window 1 Delay	5 seconds	Yes
Join Rx Window 2 Delay	6 seconds	Yes
Rx Window 2 Frequency	923.3MHz	Yes

Rx Window 2 Datarate	DR8	Yes
Default Channel Configurations	<CH No>, <Freq (MHz)>, <DRmax>, <DRmin> 0, 902.3, 3, 0 1, 902.5, 3, 0 2, 902.7, 3, 0 3, 902.9, 3, 0 4, 903.1, 3, 0 5, 903.3, 3, 0 6, 903.5, 3, 0 7, 903.7, 3, 0	Yes
Maximum Possible Number of Channels	72	No

Table 4-5: HXC900 Data Rate Table

Tx / Rx	Datarate	Spreading Factor	Payload Size (Bytes)	Bandwidth	Bit/s
Tx	DR0	SF10	11	125KHz	980
	DR1	SF9	53		1760
	DR2	SF8	125		3125
	DR3	SF7	242		5470
	DR4	SF8	242	500KHz	12500
Rx	DR8	SF12	53	500KHz	980
	DR9	SF11	129		1760
	DR10	SF10	242		3900
	DR11	SF9	242		7000
	DR12	SF8	242		12500
	DR13	SF7	242		21900

Table 4-6: HXC900 Power Table

Tx Power	Configuration (dB)
Tx5	20
Tx6	18
Tx7	16
Tx8	14
Tx9	12
Tx10	10

5 Pin Definitions

5.1 Pin Layout

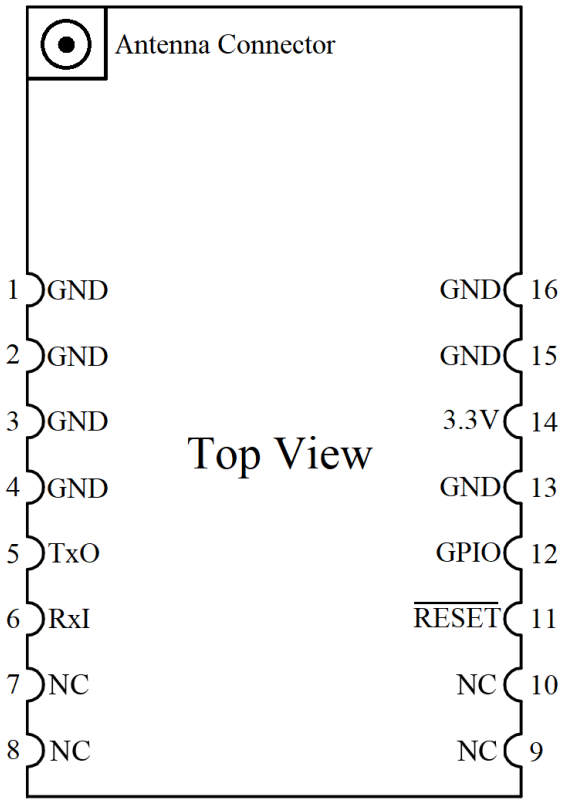


Figure 5-1: Pin Layout of HXC Client Module

5.2 Pin Descriptions

Table 5-1: Pin description of HXC Client Module

Pin No.	Pin Name	Function
1, 2, 3, 4, 13, 15, 16	GND	Power Ground
5	TxO	HXC Tx-Out Pin
6	RxI	HXC Rx-In Pin
7, 8, 9, 10	NC	No Connection
11	$\overline{\text{RESET}}$	Active low reset pin of HXC
12	GPIO	HXC GPIO for future use
14	3.3V	Power VCC

Note: The pins are 3.3V tolerant and not 5V tolerant. For 5V I/O use a logic level converter (such as this).

6 Connection Diagram

The figures below show some possible ways a user can interface an HXC client module.

6.1 Interfacing with a Microcontroller

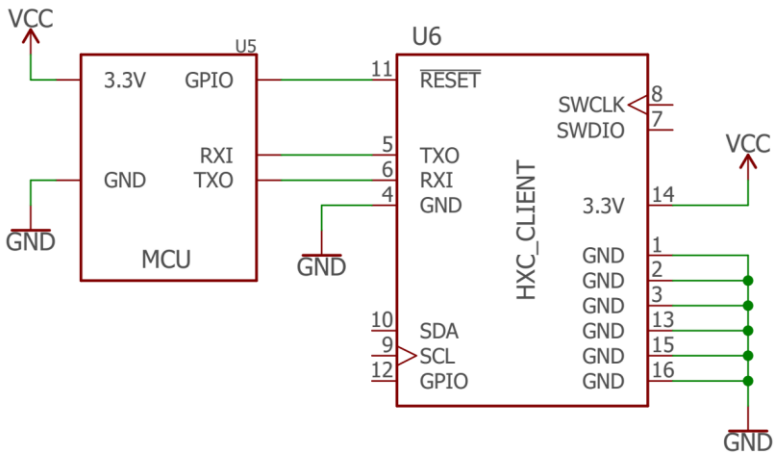


Figure 6-1: Interface between MCU and HXC Client

This is the minimal connection needed to interface with an MCU. Users will need to connect Rx and Tx of MCU with the `TXO` and `RXI` of the HXC Client, respectively. A GPIO pin to $\overline{\text{RESET}}$ pin connection is needed to reset the HXC client module. Check out our Embedded API to see how to properly initialize HXC Clients.

Note: The /RESET pin (Pin 11) is pulled-up to 3.3V (Pin 14) by 47Kohm. TxO (Pin 5) and RxI (Pin 6) are internally pulled up to 3.3V by the ARM processor.

6.2 Interfacing with a USB-to-Serial Module

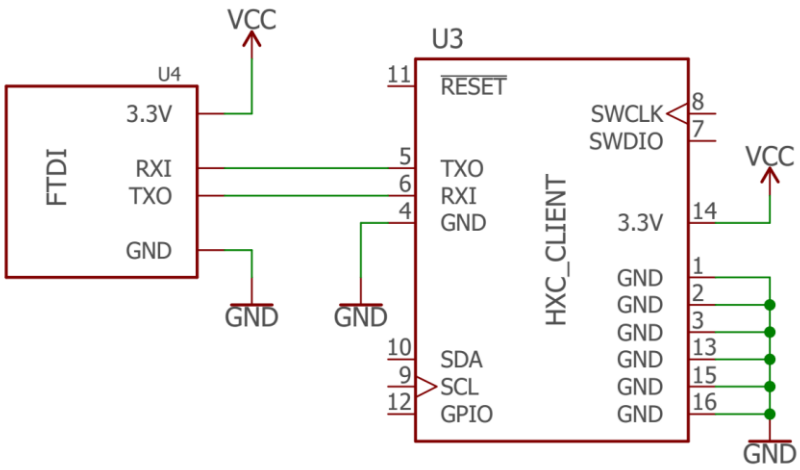
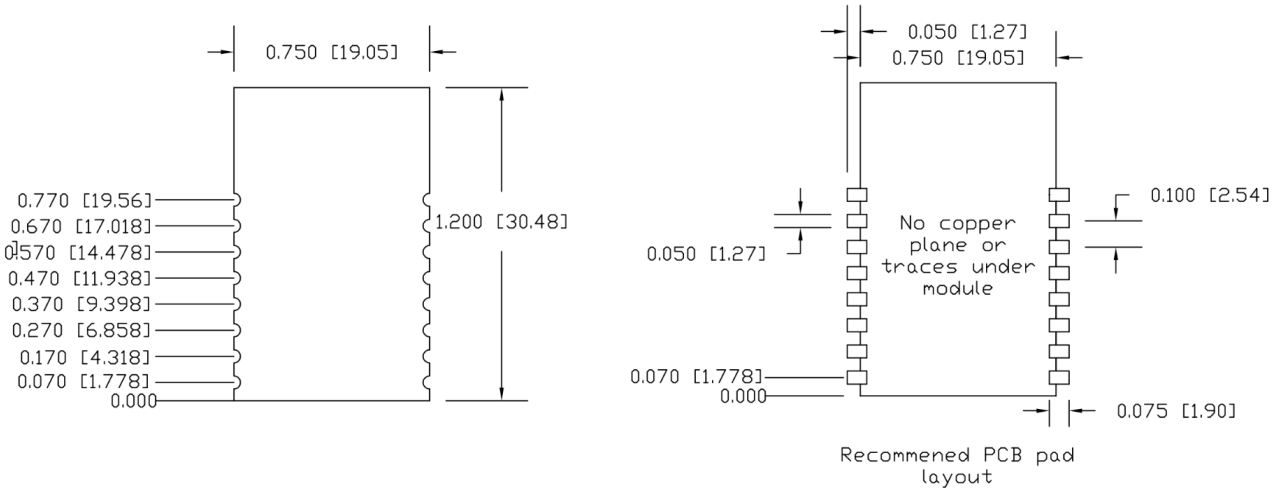


Figure 6-2: Interface between HXC Client and a USB-to-Serial Module

This is the minimal connection needed to interface with a USB-to-Serial converter (e.g. FTDI FT232RL). Our HXC Client Expansion Board can be used with Arduino and STM32 Nucleo boards that will provide a convenient USB-to-Serial interface that you can directly connect to your PC/Laptop. Users can use any terminal emulator software to directly execute AT commands. Appendix I provides an example of a 3rd party terminal emulator.

7 Package Information



Measuremnt in inches [millimeters]

Note: No copper plane, traces or oscillator immediately under the module. For example, if the module is placed on a top-layer then no traces on the top-layer under the module.

8 AT Command Reference

8.1 Overview

The AT command set is a standard developed by "Hayes" to manage modems. The HXC series modules can be managed using a simple set of ASCII AT commands over a serial UART interface. The HXC device is a slave device and must be connected to a UART master device.

8.2 Module Interface

The HXC contains a low-speed UART that can be connected to a host system such as an MCU, a serial port or a USB-to-Serial interface. The default configuration parameters for the serial interface are:

```
Baud rate: 9600
Data: 8 bits
Parity: none
Stop: 1 bit
Flow control: none
```

Appendix I describes how to set up a terminal emulator with these configurations. These settings are non-modifiable.

8.3 Conventions

- Commands are case insensitive.
- All commands have a response.
- Command length can never exceed 256 characters or bytes.
- All AT commands must end with <CR><LF>, where <CR> is carriage return and <LF> is newline/linefeed.
- Note that each line preceded by '>>>' in the examples below is the one provided by the client module to host.

8.4 Introductory Message

After a power-on-reset, the module will respond with the Hardware model number along with the firmware version and an 'OK'. A software reset will also print these messages. The example below shows the introductory message of an HXC900 module.

```
>>>HXC900 AT Interface
>>>FW Ver: 1.18
>>>OK
```

8.5 Command Syntax

All AT commands start with the standard prefix "AT" and then "+xxx" with '+xxx' denoting the specific command. For example:

AT+VER=?<CR><LF>

Here "+VER" is the command. The response (<value> or <status>) of the commands will be in the format like: "Return value<CR><LF>" and/or "Status<CR><LF>". For example, the response of "AT+VER=?" will be:

>>>1.18<CR><LF>
>>>OK<CR><LF>

There are four command behaviors:

- **Help Command: AT+XXX?**
  This provides a short help of the given command, for example:
  AT+DEVEUI?<CR><LF>
  >>>Config Device EUI<CR><LF>
  >>>OK<CR><LF>

- **Run Command: AT+XXX**
  This is used to run a command, such as:
  AT+RESET<CR><LF>
  >>>HXC900 AT Interface<CR><LF>
  >>>FW Ver: 1.18<CR><LF>
  >>>OK<CR><LF>

- **Get Command: AT+XXX=?**
  This is used to get the value of a given command, for example, '**AT+NJS=?**' will provide JOIN status.
  AT+NJS=?<CR><LF>
  >>>NOT JOINED<CR><LF>
  >>>OK<CR><LF>

- **Set Command: AT+XXX=<value>**
  This command is used to set a parameter, for example:
  AT+NJM=OTAA<CR><LF>
  >>>OK<CR><LF>

Note: Not all HXC commands have all four behaviors. The details of a specific AT command can be found in section 8.7 AT Commands.

8.6 Status and Error List

Every command returns a `Status` string with a `<CR>` and `<LF>` after the `Status` string. The possible statuses for the HXC Client module are:

- **OK**: Successful command execution.
- **AT_ERROR**: Generic error.
- **AT_PARAM_ERROR**: AT command's parameter related issues.
- **AT_BUSY_ERROR**: LoRa® network is busy, so the command could not be completed.
- **AT_PARAM_OVERFLOW**: The command length is too long.
- **AT_PAYLOAD_SIZE_ERROR**: The payload size of the LoRa packet is above the limit for the current datarate.
- **AT_NO_CHANNEL_ERROR:** No channel is available for uplink transmission.
- **AT_NO_NETWORK_JOINED**: LoRaWAN network has not been joined yet.
- **AT_INVALID_MODE**: The HXC client is in the wrong mode for current AT command (LoRa/LoRaWAN).
- **AT_RX_ERROR**: Error during the reception of the command over UART.

Users can refer to this list to understand what is happening inside HXC client when they receive errors. More details of statuses and error codes are described below.

8.6.1 AT_ERROR

There are several reasons users can receive 'AT_ERROR' responses:

- If a command does not end with '<CR><LF>'. For example:

  AT+VER=?<LF>
  >>>AT_ERROR<CR><LF>

- If a command does not start with 'AT'. For example:

  +VER=?<CR><LF>
  >>>AT_ERROR<CR><LF>

- If the user tries to use a command behavior that is not available (Refer to section 8.7 for possible AT command behaviors). For example, User cannot set/change the version of HXC Module.

  AT+VER=1.0.0<CR><LF>
  >>>AT_ERROR<CR><LF>

8.6.2 AT_PARAM_ERROR

Users can receive this error for AT command argument/parameter related cases. Such as:

- If the number of parameters is invalid. For example, Rx2 window setup command requires two parameters: frequency and data rate.

  AT+RX2WND=471200000<CR><LF>
  >>>AT_PARAM_ERROR<CR><LF>

- If the content of the parameter is invalid. For example, Network Join Mode can be either 'ABP' or 'OTAA'.

  AT+NJM=OTHER<CR><LF>
  >>>AT_PARAM_ERROR<CR><LF>

- If the command format is not right. For example, the command format of sending a packet over LoRaWAN is: (if you use a comma instead of a semicolon)

  AT+SEND=<ACK>,<PORT>:<DATA><CR><LF>
  AT+SEND=0,2,Hello World!<CR><LF>
  >>>AT_PARAM_ERROR<CR><LF>

8.6.3 AT_BUSY_ERROR

Users can receive this error for the following reasons:

- If the user initiates a transmission during an ongoing or scheduled transmission. For example, executing 'AT+SEND', right after another 'AT+SEND' command before receiving the transmission done confirmation message ('txDone').

  AT+SENDB=0,2:ABCD1234<CR><LF>
  >>>OK<CR><LF>
  AT+SENDB=0,2:FEED74ECA7<CR><LF>
  >>>AT_BUSY_ERROR<CR><LF>

- If the user tries to change any parameter that the ongoing transmission is using. For example, changing DEVEUI (AT+DEVEUI), right after 'AT+SEND' command before transmission is done ('txDone') will result in this error.

  AT+SENDB=1,100:BA134A7B1ED0<CR><LF>
  >>>OK<CR><LF>
  AT+DEVEUI=DE0120BE0EF0FACE<CR><LF>
  >>>AT_BUSY_ERROR<CR><LF>

8.6.4 AT_PAYLOAD_SIZE_ERROR

Users will receive this error if the payload size of an uplink transmission is above the limit. The payload size varies by datarate and by region. For example, on US915 region, DR_0 datarate allows a payload of 11 bytes. The example below sends 12 bytes.

AT+SENDB=0,2:1CEC00100CA700FACE0FF000<CR><LF>
>>>AT_PAYLAOD_SIZE_ERROR<CR><LF>

Check Table 4-2 and Table 4-5 for payload sizes for corresponding data rate.

8.6.5 AT_PARAM_OVERFLOW

Users will receive this error if the length of the command exceeds 256 bytes. For example:

```
AT+SENDB=0,100:A00DEAD00BEEF00BEDDED00A00DEFACED00BABE0000000000
0DEADCA70000FEED0F00D00005AD0FACE00000FA570000BA5EBA1100000000f0
05ba110000DEADBEA70000CA55E77E000000ddba11000000BBC0000b1ade0123
CAB0000CA5CADE00000C1A000AD10500123404070C0FFEE00001CE01CE0BABE0
>>>AT_PARAM_OVERFLOW<CR><LF>
```

8.6.6 AT_NO_NETWORK_JOINED

Trying to send data without joining the network first (either in ABP or OTAA mode), will result in this error. For example:

```
AT+RESET
>>>HXC900 AT Interface<CR><LF>
>>>FW Ver: 1.18<CR><LF>
>>>OK<CR><LF>
AT+SENDB=0,2:C0C033C0FFEE<CR><LF>
>>>AT_NO_NETWORK_JOINED<CR><LF>
```

8.6.7 AT_NO_CHANNEL_ERROR

Users can receive this error during any kind of uplink transmission. In the EU868 region it can be for duty cycle restriction. For other regions it will be due to datarate. If users use a datarate that none of the channel supports, they will end up with this error. For example, HXC900 has these default channels:

```
AT+CH=?<CR><LF>
>>>0,902300000,3,0<CR><LF>
>>>1,902500000,3,0<CR><LF>
>>>2,902700000,3,0<CR><LF>
>>>3,902900000,3,0<CR><LF>
>>>4,903100000,3,0<CR><LF>
>>>5,903300000,3,0<CR><LF>
>>>6,903500000,3,0<CR><LF>
>>>7,903700000,3,0<CR><LF>
>>>OK<CR><LF>
AT+DR=4<CR><LF>
>>>OK<CR><LF>
AT+SENDB=0,2:C0C033C001<CR><LF>
>>>AT_NO_CHANNEL_ERROR<CR><LF>
```

8.6.8 AT_INVALID_MODE

The HXC Client modules support both LoRaWAN protocol and LoRa only radio communication. Some AT commands are dependent on AT+MODE setting. The table below

shows what commands are dependent on AT+MODE setting. Failure to follow the table will result in AT_INVALID_MODE error.

Table 8-1: List of Commands dependent on MODE setting

AT+MODE=LORA	AT+MODE=LORAWAN	
AT+TX	AT+CLASS	AT+JOIN
AT+RX	AT+DEVEUI	AT+SEND
	AT+APPEUI	AT+SENDB
	AT+APPKEY	AT+MJOIN

8.6.9 AT_RX_ERROR

Users can receive this error for either Parity error, Frame error, Overrun error or Noise error during UART communication.

More details on each command description and examples are described in the next part of this section.

8.7 AT commands

To facilitate the description, all <CR><LF> is intentionally omitted for all the examples listed from here on, unless otherwise specified.

8.7.1 General Commands

8.7.1.1 AT: Attention

This command is used to check if the connection between module and host is OK. *Run* command will respond with an 'OK'. *Help* command provides short help for all the supported commands. There aren't any *Get* or *Set* command for AT. Executing those commands will return AT_ERROR.

```
AT
>>>OK
AT?
>>>AT+<CMD>?       : Help on <CMD>
>>>AT+<CMD>        : Run <CMD>
>>>AT+<CMD>=<value>: Set the value
>>>AT+<CMD>=?      : Get the value
… … …
… … …
… … …
>>>RX: Receive LoRa Packet
>>>RADIO: Get Radio IC Version
>>>MODE: Switch Between LoRa & LoRaWAN
>>>OK
AT=?
>>>AT_ERROR
```

8.7.1.2 AT+RESET: Reset HXC Client Module

This command triggers a CPU reset of the HXC client module. *Run* command will respond with the welcome message following an 'OK'. There aren't any *Get* or *Set* command.

```
AT+RESET?
>>>Reset the Client
>>>OK
AT+RESET
>>>HXC900 AT Interface
>>>FW Ver: 1.18
>>>OK
```

8.7.1.3 AT+FD: Reset to Factory Defaults

This command resets all AT parameters to their default value, erases all saved parameters in EEPROM and triggers a CPU reset of the HXC client module. There aren't any *Get* or *Set* command for AT+FD.

```
AT+FD?
>>>Reset to Factory Default
>>>OK
```

```
AT+FD
>>>HXC900 AT Interface
>>>FW Ver: 1.18
>>>OK
```

Users can receive a busy error if the EEPROM is busy or not accessible during the command execution.

```
AT+FD
>>>AT_BUSY_ERROR
```

8.7.2 LoRaWAN Keys

This section provides commands specific to the keys and identifiers used in the HXC client module that is required for connecting to a LoRaWAN network. Different keys are required for different authentication schemes.

'Over-The-Air-Activation' (OTAA) requires 3 keys to authenticate against the LoRaWAN Network Server: `APPEUI`, `DEVEUI`, and `APPKEY`. The Network Server dynamically provides the remaining keys when the network is joined.

'Activation-by-Personalization' (ABP) requires 6 keys to be statically configured on the module. All 6 keys must be configured for the module to successfully encrypt and send or receive a message. The keys are `APPEUI`, `DEVEUI`, `APPKEY`, `DEVADR`, `NWKSKEY`, and `APPSKEY`.

NOTE: ABP is a less secure connectivity option for LoRaWAN, as the keys are statically configured and are not reset at every JOIN.

Appendix II lists set of AT commands to use for OTAA and ABP authentication schemes, respectively.

8.7.2.1 AT+DEVEUI: Device EUI

The device EUI is an EUI-64 value that is unique to the device. All HXC client modules contain a globally unique EUI-64 address. This EUI can be used by configuring the DEVEUI value to "`AUTO`" (default). Users can also assign any other value as DEVEUI. It takes 16digit (8byte) hex value. Hex values are case-insensitive. Any change of DEVEUI using *Set command* is saved in module's EEPROM and can be retained after a Reset.

Format: `AT+DEVEUI=<Param>`

```
AT+DEVEUI?
>>>Config Device EUI
>>>OK
AT+DEVEUI=AUTO
>>>OK
AT+DEVEUI=?
>>>10:01:13:1B:10:10:1F:15    //Globally unique ID, different for every other device
>>>OK
AT+DEVEUI=8899aabbccddeeff  // Setting a custom DEVEUI
>>>OK
```

Users can receive `AT_PARAM_ERROR` if the parameter has less than 8 bytes.

```
AT+DEVEUI=8899aabbccddee
>>>AT_PARAM_ERROR
```

Users can receive `AT_BUSY_ERROR` if the command is used during a LoRa transmission*. The user will have to wait until the transmission is done.

*LoRa transmission = Join procedure, single packet uplink transmission or continuous transmission

```
AT+DEVEUI=11de45ad0000face
>>>AT_BUSY_ERROR
```

Users can receive AT_BUSY_ERROR during **AUTO** command if the processor fails to read the EUI64 bytes.

```
AT+DEVEUI=AUTO
>>>AT_BUSY_ERROR
```

8.7.2.2 AT+DEVADR: Device Address

This command allows users to access the device address. Default value is 00:00:00:00. OTAA authentication scheme will provide the DEVADR during JOIN procedure. During ABP, the user will have to define the DEVADR. The command takes 8digit (4byte) hex value. Hex values are case-insensitive. Any change of DEVADR using *Set command* is saved in EEPROM.

```
AT+DEVADR?
>>>Config Device Address
>>>OK
AT+DEVADR=112233AA
>>>OK
```

Users can receive AT_PARAM_ERROR if the parameter has less than 4 bytes.

```
AT+DEVADR=112233A
>>>AT_PARAM_ERROR
```

Users can receive AT_BUSY_ERROR if the command is used during a LoRa transmission*. The user will have to wait until the transmission is done.

8.7.2.3 AT+APPEUI: Application Identifier

This command allows users to access the global application identifier. The default value is 0. The command takes 16digit (8byte) hex value. Hex values are case-insensitive. Any change of APPEUI using *Set command* is saved in EEPROM and can be retained after a Reset.

```
AT+APPEUI?
>>>Config Application EUI
>>>OK
AT+APPEUI=ca7f00d0000666
>>>OK
```

Users can receive AT_PARAM_ERROR if the parameter has less than 8 bytes.

```
AT+APPEUI=11223344556677
>>>AT_PARAM_ERROR
```

Users can receive AT_BUSY_ERROR if the command is used during LoRa transmission*.

*LoRa transmission = Join procedure, single packet uplink transmission or continuous transmission

8.7.2.4 AT+APPKEY: Application Key

This command allows users to set the application session key. For security reasons, this value is a **write-only** parameter. The command takes 32digit (16byte) hex value. Hex values are case-insensitive. The default value is 0. Any change of APPKEY using *Set command* is saved in EEPROM and automatically set as APPKEY after a reset.

```
AT+APPKEY?
>>>Set Application Key
>>>OK
AT+APPKEY=00112233445566778899aabbccddeeff
>>>OK
```

Users can receive AT_ERROR if they try to read the key.

```
AT+APPKEY=?
>>>AT_ERROR
```

Users can receive AT_PARAM_ERROR if the parameter has less than 16 bytes.

```
AT+APPKEY=112233445566778899
>>>AT_PARAM_ERROR
```

Users can receive AT_BUSY_ERROR if the command is used during LoRa transmission*.

8.7.2.5 AT+APPSKEY: Application Session Key

This command allows users to set the application session key. For security reasons, this value is a **write-only** parameter. The default value is 0. Using the OTAA scheme will set the APPSKEY during JOIN procedure. The command takes 32digit (16byte) hex value. Hex values are case-insensitive. Any change of APPSKEY using *Set command* is saved in EEPROM and automatically set as APPSKEY after a reset.

```
AT+APPSKEY?
>>>Set Application Session Key
>>>OK
AT+APPSKEY=00112233445566778899aabbccddeeff
>>>OK
```

Users can receive AT_ERROR if they try to read the key.

```
AT+APPSKEY=?
>>>AT_ERROR
```

Users can receive AT_PARAM_ERROR if they supply less than 16 bytes. Users can receive AT_BUSY_ERROR if this command is used during LoRa transmission*. Users will have to wait until the transmission is done.

*LoRa transmission = Join procedure, single packet uplink transmission or continuous transmission

8.7.2.6 AT+NWKSKEY: Network Session Key

This command allows users to set the network session key. For security reasons, this value is a **write-only** parameter. The default value is 0. Using the OTAA scheme will set the APPSKEY during JOIN procedure. The command takes 32digit (16byte) hex value. Hex values are case-insensitive. Any change of NWKSKEY using *Set command* is saved in EEPROM and automatically set as NWKSKEY after a reset.

```
AT+NWKSKEY?
>>>Set Network Session Key
>>>OK
AT+NWKSKEY=00112233445566778899aabbccddeeff
>>>OK
```

Users can receive `AT_ERROR` if they try to read the key.

```
AT+NWKSKEY=?
>>>AT_ERROR
```

Users can receive `AT_PARAM_ERROR` if the parameter has less than 32 digits (16 bytes).

```
AT+NWKSKEY=112233445566778899
>>>AT_PARAM_ERROR
```

Users can receive `AT_BUSY_ERROR` if the command is used during LoRa transmission*. Users will have to wait until the transmission is done.

*LoRa transmission = Join procedure, single packet uplink transmission or continuous transmission

8.7.3 Join, Sending and Receiving Data

To send data, the HXC module must have the correct keys loaded and joined a network.

8.7.3.1 AT+NJM: LoRa® Network Join Mode

This command allows users to access the network join mode. The input parameters are either 'ABP' (Activation by Personalization) or 'OTAA' (Over-the-Air Activation). The default is OTAA. Any change of NJM using *Set command* is saved in EEPROM.

```
AT+NJM?
>>>Config Network Join Mode
>>>OK
AT+NJM=?
>>>OTAA
>>>OK
AT+NJM=ABP
>>>OK
```

The user can receive `AT_PARAM_ERROR` if the parameter is neither ABP nor OTAA.

```
AT+NJM=5
>>>AT_PARAM_ERROR
```

8.7.3.2 AT+JOIN: Join LoRa® Network

This command sends a join request to the network. If the join mode is OTAA (Over-the-Air Activation), this command will send the Join Request and wait for JN1DL milliseconds for RX1 window and JN2DL milliseconds for the RX2 window to listen for a 'Join Accept' packet from the network server. Also note, Region-CN470 join request will use datarate set by `AT+DR` (default DR0). Region-US915 join request will use DR0 by default and every 9th join request will use DR4.

The client module takes one attempt to join with a network server and then go back to sleep. During ABP join mode, although the Join Request isn't necessary, users will have to use this command to update the state of the client module. An example of how to use this command can be found in Appendix II.

Note: Users can send a join request anytime they want, even if the device is already joined, to update the dynamic security session keys.

```
AT+JOIN?
>>>Join Network
>>>OK
```

// Case: OTAA – Successfully joined

```
AT+JOIN
>>>OK
>>>txDone
>>>rxDone
>>>JOINED
```

// Case: OTAA – Failed to join (in Class A)

```
AT+JOIN
>>>OK
>>>txDone
>>>rx1Timeout
>>>rx2Timeout
```

```
//Case: ABP Join Scheme
AT+JOIN
>>>OK
>>>JOINED
```

Users can receive AT_BUSY_ERROR if the command is used during a LoRa transmission*. Users will have to wait until the transmission is done.

```
AT+JOIN
>>>AT_BUSY_ERROR
```

Users can receive AT_INVALID_MODE if the MODE is not LORAWAN.

```
AT+JOIN
>>>AT_INVALID_MODE
AT+MODE=?
>>>LORA
>>>OK
```

The status of the Join Request can be verified with AT+NJS.

8.7.3.3 AT+NJS: LoRa® Network Join Status

This command provides users the join status of the LoRa®. It is a read-only command. The response can be either JOINED or NOT JOINED. The command provides the last join request status.

```
AT+NJS?
>>>Get Network Join Status
>>>OK
AT+JOIN
>>>OK
>>>txDone
>>>rxDone
>>>JOINED
AT+NJS=?
>>>JOINED
>>>OK
AT+JOIN              // Sending join request again to update dynamic session keys
>>>OK
>>>txDone
>>>rx1Timeout
>>>rx2Timeout    // Device failed to connect during 2nd join request
AT+NJS=?
>>>NOT JOINED    // NJS provides the last join request status
>>>OK
```

*LoRa transmission = Join procedure, single packet uplink transmission or continuous transmission

8.7.3.4 AT+SEND: Send ASCII Bytes

This command provides a way to initiate an uplink transmission using ASCII bytes as payload on a dedicated port and with acknowledgment configuration (Confirmed/ Unconfirmed). The confirmed message takes 8 attempts to receive ACK from NS before it goes back to sleep. The uplink transmission will use default datarate (section 4.5) and can be updated using AT+DR command.

If the command is successfully executed, the module will respond with 'OK'. Then it will print txDone if the transmission is successful. After 'Receive Window 1 Delay', the module will wake up from sleep and will enable the receive channel for any downlink packet from a network server (e.g. X-ON). If there aren't any downlink packets module will print rx1Timeout and will go back to sleep. Same applies to the 2nd Rx window. If there is any downlink packet, instead of 'rxTimeout' HXC client will print 'rxDone'.

Note: It is advisable to wait till rx2Timeout for Class A and rx1Timeout for Class C, before sending the next packet. Class C devices always listen on Receive window 2, hence no rx2Timeout for Class C devices.

Format: AT+SEND=<AckReq>,<Port>:<ASCII bytes>
<AckReq> = 0(:Unconfirmed) or 1(:Confirmed)
<Port>= 1 to 223

AT+SEND?
>>>Send ASCII Uplink
>>>OK

//Case: Class A Unconfirmed uplink
AT+SEND=0,2:hello
>>>OK
>>>txDone
>>>rx1Timeout
>>>rx2Timeout

//Case: Class C Unconfirmed uplink
AT+SEND=0,2:World!
>>>OK
>>>txDone
>>>rx1Timeout

//Case: Confirmed uplink with ACK received from NS
AT+SEND=1,2:hello
>>>OK
>>>txDone
>>>rxDone // Users can check the ACK status using AT+CFS command

//Case: Class A Confirmed uplink with no ACK from NS
AT+SEND=1,2:mars!
>>>OK
>>>txDone // 1st attempt
>>>rx1Timeout
>>>rx2Timeout

```
… … …
>>>txDone          // 8th attempt
>>>rx1Timeout
>>>rx2Timeout
```

Users can receive AT_BUSY_ERROR if the command is used during a LoRa transmission. Users will have to wait till the transmission is done.

```
AT+SEND=0,2:hello
>>>AT_BUSY_ERROR
```

Users can receive AT_NO_NETWORK_JOINED if the command is used before joining a network (using AT+JOIN), for both OTAA and ABP join mode.

```
AT+SEND=0,2:world!
>>>AT_NO_NETWORK_JOINED
```

Users can receive AT_INVALID_MODE if the MODE is not LORAWAN.

```
AT+SEND=0,2:hello
>>>AT_INVALID_MODE
AT+MODE=?
>>>LORA
>>>OK
```

Users can receive AT_PAYLOAD_SIZE_ERROR if the payload size is above the limit. For example, on US915 region, with DR_0 datarate, payload size can't be more than 11 bytes.

```
AT+SEND=0,2:how big is the payload?
>>>AT_PAYLOAD_SIZE_ERROR
```

Users can receive AT_PARAM_ERROR if the command format is not right or the <AckReq> and/or <Port> values are out of range.

8.7.3.5 AT+SENDB: Send Hexadecimal Bytes

This command provides a way to send hexadecimal bytes on a dedicated port, with acknowledgment configuration (Confirmed/Unconfirmed). The confirmed message takes 8 attempts to receive ACK from Network Server before it goes back to sleep. The uplink transmission will use default datarate (section 4.5) and can be updated using AT+DR command.

Each byte of the hex data consists of two digits. Hence, the length of the binary data is always even. For example, if the users want to send 0x1, they will have to send "01" instead of "1". Hex digits are case-insensitive.

If the command execution is successful, the module will respond with 'OK'. Then it will print txDone after the transmission is successful. After 'Receive Window 1 Delay', the module will wake up from sleep and will enable the receive channel for any downlink packet from NS. If there aren't any downlink packet module will print rx1Timeout. Same applies to the second receive window. If there is any downlink packet, instead of 'rxTimeout' HXC client will print 'rxDone'.

Note: It is advisable to wait till rx2Timeout for Class A and rx1Timeout for Class C, before sending the next packet. Class C devices always listen on Receive window 2, hence no rx2Timeout for Class C devices.

Format: AT+SENDB=<AckReq>,<Port>:<Hexadecimal bytes>
<AckReq> = 0(:Unconfirmed) or 1(:Confirmed)
<Port>= 1 to 223

AT+SENDB?
>>>Send Hexadecimal Uplink
>>>OK

//Case: Class A unconfirmed uplink
AT+SENDB=0,2:C0C0C0FFEE
>>>OK
>>>txDone
>>>rx1Timeout
>>>rx2Timeout

//Case: Class C unconfirmed uplink
AT+SENDB=0,2:C001C0FFEE
>>>OK
>>>txDone
>>>rx1Timeout

//Case: Confirmed uplink with ACK from NS
AT+SENDB=1,2:5AD0FACE
>>>OK
>>>txDone
>>>rxDone

//Case: Class C confirmed uplink with no ACK from NS
AT+SENDB=1,2:1270F034
>>>OK
>>>txDone // 1st attempt
>>>rx1Timeout
… … … … … … … …
>>>txDone // 8th attempt
>>>rx1Timeout

Users can receive AT_BUSY_ERROR if the command is used during a LoRa transmission.

AT+SENDB=0,2:407F00D9
>>>AT_BUSY_ERROR

Users can receive AT_INVALID_MODE if the MODE is not LORAWAN.

```
AT+SENDB=0,2:3F4C70FF
>>>AT_INVALID_MODE
AT+MODE=?
>>>LORA
>>>OK
```

Users can receive AT_NO_NETWORK_JOINED if the command is executed before joining a network, for both OTAA and ABP join mode.

```
AT+SENDB=0,2:CA70F00D
>>>AT_NO_NETWORK_JOINED
```

Users can receive AT_PAYLOAD_SIZE_ERROR if the payload size is above the limit. For example, on Redion-US915, with DR_0 datarate, payload size can't be more than 11 bytes.

```
AT+SENDB=0,2:A0C0010CA700BED12300A0B0
>>>AT_PAYLOAD_SIZE_ERROR
```

Users can receive AT_PARAM_ERROR if the command format is not right, <AckReq> or <Port> values are out of range or if the payload doesn't have an even number of digits.

8.7.3.6 Downlink Packet

There are two ways to get downlink packets. This section explains the first method. The second method is explained here.

Users will be notified about any downlink packet through UART as soon as the packet is received. The payload is printed using hexadecimal bytes along with the port number and NS ACK flag i.e. if this is an unconfirmed or confirmed downlink packet. The format is:

```
>>>rxDone<CR><LF>
>>>rxdata:<NsAckFlag>,<Port>,<Payload><CR><LF>
```

<NsAckFlag> = 0: No Ack required from client; 1: NS requested an ACK from client

HXC client module will **acknowledge automatically** during the **next regular LoRa packet transmission** if the NS requested an ACK. It's up to the application layer/user **when** to send a packet. 'NsAckFlag' just informs the application layer/user that NS requested an acknowledgment so that user can schedule a transmission.

In Class A, the receive windows are open after an uplink transmission (using AT+SEND/ SENDB). So, the downlink packet (if there are any) will be available right after an uplink transmission. During Class C, data can be received anytime on receive window 2. This method explained above is particularly useful for Class C because the application device can receive downlink packets at any time. AT+RSSI and AT+SNR can be used after a downlink packet is received to check the quality of the message.

Example:
Downlink packet received at port 2 on Class A after an uplink transmission:

```
AT+SENDB=0,2:10666040
>>>txDone
>>>rxDone
>>>rxdata:0,2,C0010CA7
```

Downlink packet received at port 2 and NS requested for an ACK from a Class C device:

```
>>>rxDone
>>>rxdata:1,2,5AD0CA7  // Note: To receive downlink in Class C, user doesn't need an uplink

AT+SENDB=0,2:DE6666AD  // A regular LoRaWAN uplink. The acknowledgment for the
>>>txDone               // received data will be added automatically with this transmission
>>>rx1Timeout           // Note: There aren't any rx2Timeout for Class C
```

8.7.3.7 AT+RECVB: Get Last Downlink Packet

This section explains the second method to get downlink packets. This command allows the users to get the last downlink packet received. Once the payload is delivered to the user HXC Client will erase the packet from memory. A second call will yield NULL. It is a read-only command.

```
AT+RECVB?
>>>Get Last Downlink
>>>OK
```

```
//Class A Example
AT+SENDB=0,2:CA110CAB
>>>OK
>>>txDone
>>>rxDone
>>>rxData:0,2,3AD105
AT+RECVB=?
>>>0,2,3AD105
>>>OK
AT+RECVB=?
>>>OK
```

```
// Class C Example
>>>rxDone
>>>rxData:0,2,1CA5CADE
AT+RECVB=?
>>>0,2,1CA5CADE
>>>OK
AT+RECVB=?
>>>OK
```

8.7.3.8 AT+CFS: Confirm Status

This command allows the users to check the status of the last LoRaWAN packet transmission. It can be either 0(Unconfirmed) or 1(Confirmed). It is a read-only command.

```
AT+CFS?
>>>Get Ack Status of the last Uplink
>>>OK
```

```
AT+SENDB=0,2:1CE0C001  // Unconfirmed Transmission
>>>OK
>>>txDone
>>>rx1Timeout
>>>rx2Timeout
AT+CFS=?
>>>0                   // No ACK from NS
>>>OK
AT+SENDB=1,2:BA5EBA11  // Confirmed Transmission
>>>OK
>>>txDone
>>>rxDone
AT+CFS=?
>>>1                   // ACK received from NS
>>>OK
AT+SEND=1,2:99.3F      // Confirmed Transmission
>>>OK
>>>txDone              // 1st attempt
>>>rx1Timeout
>>>rx2Timeout
… … …
>>>txDone              // 8th attempt
>>>rx1Timeout
>>>rx2Timeout
AT+CFS=?
>>>0                   // No ACK from NS
>>>OK
```

8.7.4 LoRaWAN Network Management

This section provides commands that enable network management functions.

8.7.4.1 AT+CH: Add or Remove Channel

This command allows the user to add or remove channels. Channel configurations are different for different regions.

CN470 region:

HXC400 supports CN470 region. There will be eight default channels. The default channel configurations can be found in section 4.5.1. Users can add up to 16 channels. The range of channel ID is 0 to 15. All channels are configurable including default channels, but the first three channels cannot be removed. Updating channel configurations using *SET* command are saved in EEPROM.

Format:

Add Channel: `AT+CH=<Ch ID>,<Freq HZ>,<DRmax>,<DRmin>`

Remove Channel: `AT+CH=<Ch ID>,0`

```
AT+CH?
>>>Config Channel
>>>OK
AT+CH=?                          // Region-CN470Default Channels
>>>0,471500000,5,0
… … …
>>>7,472900000,5,0
>>>OK
AT+CH=8,472100000,3,0
>>>OK
AT+CH=?
>>>0,471500000,5,0
… … …
>>>7,472900000,5,0
>>>8,472100000,3,0
>>>OK
AT+CH=8,0
>>>OK
AT+CH=?
>>>0,471500000,5,0
… … …
>>>7,472900000,5,0
>>>OK
```

Users can receive `AT_PARAM_ERROR` if the command format is not right, if any of the input parameter is out of range or if user is trying to remove default channels. Frequency range is 433MHz – 510MHz. Datarate range is 0 to 5.

```
AT+CH=1,0
>>>AT_PARAM_ERROR
AT+CH=5,923000000,7,0
>>>AT_PARAM_ERROR
```

US915 region:

HXC900 supports US915 region. There will be 8 default channels. The default channel configurations are shown in section 4.5.2. Users can add up to 72 channels. The channels are configurable by group number. Each group represents 8 channels. For example, group-0 represents channel 0 to 7, group-1 represents channel 8 to 15 and so on. There are total 9 groups. All groups are configurable. You can also remove the channels by group except for group-zero (Channel 0 to 7). All channel configurations are saved in EEPROM.

The command takes a base frequency, maximum and minimum datarate. The first channel of a group will use the base frequency and the consecutive channels' frequency will be incremented by 200KHz. All the groups use 125KHz bandwidth except group-8. Group-8 represents special channels (channel 64 to 71), which uses 500KHz bandwidth. The consecutive channels' frequency of group 8 will be incremented by 1.6MHz from the base frequency.

Format:
Add Channel: `AT+CH=<Group ID>,<Base Freq HZ>,<DRmax>,<DRmin>`
Remove Channel: `AT+CH=<Group ID>,0`

```
AT+CH?
>>>Config Channel Group
>>>OK
AT+CH=?                                  // Region_US915 Default channels
>>>0,902300000,3,0
… … …
>>>7,903700000,3,0
>>>OK
AT+CH=1,912300000,3,0
>>>OK
AT+CH=?
>>>0,902300000,3,0
… … …
>>>7,903700000,3,0
>>>8,912300000,3,0
>>>9,912500000,3,0
>>>10,912700000,3,0
>>>11,912900000,3,0
>>>12,913100000,3,0
>>>13,913300000,3,0
```

```
>>>14,913500000,3,0
>>>15,913700000,3,0
>>>OK
AT+CH=1,0
>>>OK
AT+CH=?
>>>0,902300000,3,0
… … …
>>>7,903700000,3,0
>>>OK
```

Users can receive AT_PARAM_ERROR if the command format is not right, if any of the input parameter is out of range or if user is trying to remove default channels. Frequency range is 902.3MHz – 930MHz. Datarate range is 0 to 4.

```
AT+CH=0,0
>>>AT_PARAM_ERROR
AT+CH=5,900000000,7,0
>>>AT_PARAM_ERROR
```

8.7.4.2 AT+RX1FR: Receive Window 1 Frequency (HXC400 only)

This command allows users to read and modify the receive-window-1 frequency of a channel. The default receive-window-1 frequency of a channel is the same as the transmit frequency of that channel. The range of applicable values is 433MHz to 510MHz. Data rates are the same as channel data rates. These configurations are saved in EEPROM.

```
AT+RX1FR?
>>>Get or Set Downlink RX1 Freq
>>>OK
AT+RX1FR=?                  // CN470 Region Default Downlink Freq
>>>0,471500000
… … …
>>>7,472900000
>>>OK
AT+RX1FR=2,472100000
>>>OK
AT+RX1FR=?
>>>0,471500000
>>>1,471700000
>>>2,472100000
… … …
>>>7,472900000

>>>OK
```

Users can receive AT_PARAM_ERROR if the command format is not right, any of the input parameters is out of range or if the user tries to change the downlink frequency of a non-

existing channel. For example, if you have 3 channels and try to add a downlink frequency for a fourth channel.

```
AT+CH=?
>>>0,471500000,5,0
>>>1,471700000,5,0
>>>2,471900000,5,0
>>>OK
AT+RX1FR=3,471900000
>>>AT_PARAM_ERROR
```

8.7.4.3 AT+ADR: Adaptive Datarate

This command allows users to access the adaptive data rate parameter. The command will take either 0(:Disable) or 1(:Enable). The default value of the ADR is 0. This setting is saved in EEPROM and can be retain after a reset.

```
AT+ADR?
>>>Config Adaptive Data Rate
>>>OK
AT+ADR=?
>>>0                  // Disabled
>>>OK
AT+ADR=1            // Enable ADR
>>>OK
```

Users can receive `AT_PARAM_ERROR` if the input parameter neither 0 nor 1.

```
AT+ADR=2
>>>AT_PARAM_ERROR
```

8.7.4.4 AT+CLASS: LoRaWAN™ Class

This command allows the user to access the LoRaWAN™ Classes. Currently, HXC Client Module supports Class A and C (Class B is not supported yet). The default is Class A. This setting is saved in EEPROM. More about the classes can be found in Section 4.

```
AT+CLASS?
>>>Config Device Class
>>>OK
AT+CLASS=?
>>>A
>>>OK
AT+CLASS=C
>>>OK
AT+CLASS=B
>>>AT_PARAM_ERROR
```

8.7.4.5 AT+DCS: Duty Cycle settings (European Region only)

This command allows the user to access the duty cycle parameter. This command is for testing only. Valid input parameters are 0(:Disable) and 1(:Enable). The default is 0.

```
AT+DCS?
>>>Config ETSI Duty Cycle
>>>OK
AT+DCS=?
>>>0                 // Disabled
>>>OK
AT+DCS=1            // Enable Duty Cycle
>>>OK
```

8.7.4.6 AT+DR: Data Rate

This command allows the user to access the data rate. The data rates vary by region. Data Rate may be manually set only when ADR is disabled. The default value varies by region (check section 4.5). Updating the datarate will affect the uplink transmission and maybe join request (depends on region). This setting is saved in EEPROM.

```
AT+DR?
>>>Config Uplink Datarate
>>>OK
AT+DR=?
>>>0                          // Default DR on Region-CN470
>>>OK
AT+DR=1
>>>OK
```

The user can receive `AT_PARAM_ERROR` if the input parameter is out of range. The range varies by region (check section 4.5).

```
AT+DR=5
>>>AT_PARAM_ERROR       // Region_US915
```

8.7.4.7 AT+FCU: Uplink Frame Counter

This command allows the user to access the uplink frame counter. The counter increments each time a successful uplink transmission occurs (not including Join Request).

```
AT+FCU?
>>>Config Uplink Frame Counter
>>>OK
AT+FCU=?
>>>0
>>>OK
AT+FCU=1
>>>OK
```

8.7.4.8 AT+FCD: Downlink Frame Counter

This command allows the user to access the downlink frame counter. The counter increments each time a downlink packet is received.

```
AT+FCD?
>>>Config Downlink Frame Counter
>>>OK
AT+FCD=?
>>>0
>>>OK
AT+FCD=10
>>>OK
```

8.7.4.9 AT+JN1DL: Join Accept Delay of RX window 1

This command allows the user to read and modify the join accept delay for Rx window 1, in milliseconds. The default value is 5000ms (5s).

```
AT+JN1DL?
>>>Config Join Accept Delay for RX1WND
>>>OK
AT+JN1DL=?
>>>5000
>>>OK
AT+JN1DL=1000
>>>OK
```

Users can receive `AT_BUSY_ERROR` if the command is sent during a LoRa transmission*.

```
AT+JN1DL=5000
>>>AT_BUSY_ERROR
```

8.7.4.10 AT+JN2DL: Join Accept Delay of RX window 2

This command allows the user to read and modify the join accept delay for Rx window 2, in milliseconds. The default value is 6000ms (6s).

```
AT+JN2DL?
>>>Config Accept Delay for RXWND
>>>OK
AT+JN2DL=?
>>>6000
>>>OK
AT+JN2DL=1000
>>>OK
```

Users can receive `AT_BUSY_ERROR` if the command is used during a LoRa transmission*.

*LoRa transmission = Join procedure, single packet uplink transmission or continuous transmission

8.7.4.11 AT+RX1DL: Received Window 1 Delay

This command allows users to read and modify the delay between Tx and Rx window 1, in milliseconds. The default value is 1000ms (1s).

```
AT+RX1DL?
>>>Config Delay Between TX & RX1WND
>>>OK
AT+RX1DL=?
>>>1000
>>>OK
AT+RX1DL=3000
>>>OK
```

Users can receive `AT_BUSY_ERROR` if the command is used during a LoRa transmission*.

8.7.4.12 AT+RX2DL: Received Window 2 Delay

This command allows the user to read and modify the delay between Tx and Rx window 2, in milliseconds. The default value is 2000ms (2s).

```
AT+RX2DL?
>>>Config Delay Between TX & RX2WND
>>>OK
AT+RX2DL=?
>>>2000
>>>OK
AT+RX2DL=3000
>>>OK
```

Users can receive `AT_BUSY_ERROR` if the command is sent during a LoRa transmission*.

8.7.4.13 AT+RX2WND: Configure Receive Window 2

This command allows users to access the data rate (corresponding to DR_X) and the frequency (in Hz) of receive window 2. The range of frequency and data rate varies by the region. Format: `AT+RX2WND=<Freq>,<DR>`

```
AT+RX2WND?
>>>Configure RX2 Window Freq & Datarate
>>>OK
AT+RX2WND=?
>>>923300000,8        // Region_US915
>>>OK
AT+RX2WND=923900000,8
>>>OK
```

Users can receive `AT_PARAM_ERROR` if the command format is not right or parameters are out of range. The parameter range varies by region (check section 4.5).

*LoRa transmission = Join procedure, single packet uplink transmission or continuous transmission

```
AT+RX2WND=510000000,8
>>>AT_PARAM_ERROR      // Region_CN470
```

Users can receive AT_BUSY_ERROR if the command is used during a LoRa transmission*.

Note: It is advisable to change RX2WND parameters after a successful Join. Some region doesn't allow any other RX2WND parameters other than the default during Join procedure. In that case, any change of the RX2WND prior to Join will be discarded and will be restored to default.

8.7.4.14 AT+TXP: Transmit Power

This command allows the user to read and modify the transmit power. The default value and input parameter range vary by region. Check Table 4-1 and 4-4 to see the relation between these values and dB.

```
AT+TXP?
>>>Config Transmit Power
>>>OK
AT+TXP=?
>>>5                    // Region_US915
>>>OK
AT+TXP=6
>>>OK
```

Users can receive AT_PARAM_ERROR if the input parameter is not supported by the region.

```
AT+TXP=0
>>>AT_PARAM_ERROR      // Region_US915
```

8.7.4.15 AT+PNM: Public Network Mode

This command allows users to read and modify the network mode. Valid input parameters are 0(:Private) and 1(:Public). The default is 1.

```
AT+PNM?
>>>Config Public Network Mode
>>>OK
AT+PNM=?
>>>1                    // Enabled (Default Value)
>>>OK
AT+PNM=0               // Disable i.e. Private mode
>>>OK
```

Users can receive AT_PARAM_ERROR if the input parameter is not valid.

```
AT+PNM=2
>>>AT_PARAM_ERROR
```

*LoRa transmission = Join procedure, single packet uplink transmission or continuous transmission

8.7.5 Multicast Commands

Multicast uses a different set of keys from the unicast. Multicast requires a Group ID, a Group Key, and an Application Key. Multicast requires the device to be in Class C. Users will have to send a multicast join request to join a multicast group on X-ON. Once it is joined, a regular interval keep-alive timer will keep the device in that group. Turning off the timer will take out the device from the multicast group. Multicast downlinks will use RX2 window parameters.

8.7.5.1 AT+MGROUPID: Multicast Group ID

This command allows users to set the multicast group ID. Default value is 00:00:00:00:00:00. The command takes 12digit (6bytes) hex value. Hex values are case-insensitive.

```
AT+MGROUPID?
>>>Config Multicast Group ID
>>>OK
AT+MGROUPID=ca7f00d0000666
>>>OK
```

Users can receive `AT_PARAM_ERROR` if the parameter has less than 6 bytes.

```
AT+MGROUPID=1122334455
>>>AT_PARAM_ERROR
```

Users can receive `AT_BUSY_ERROR` if the command is used during LoRa transmission*.

8.7.5.2 AT+MGROUPKEY: Multicast Group Key

This command allows users to set the multicast group key. Default value is 00:00:00:00:00:00:00:00:00:00. The command takes 20digit (10bytes) hex value. Hex values are case-insensitive.

```
AT+MGROUPKEY?
>>>Config Multicast Group Key
>>>OK
AT+MGROUPKEY=CA7F00D456A5B498C15F
>>>OK
```

Users can receive `AT_PARAM_ERROR` if the parameter has less than 6 bytes. Users can receive `AT_BUSY_ERROR` if the command is used during LoRa transmission*.

8.7.5.3 AT+MAPPKEY: Multicast Application Key

This command allows users to set the multicast application session key. For security reasons, this value is a **write-only** parameter. The command takes 16-bytes hex value. Hex values are case-insensitive. The default value is 0.

```
AT+MAPPKEY?
>>>Set Multicast Application Key
>>>OK
AT+MAPPKEY=00112233445566778899aabbccddeeff
>>>OK
```

Users can receive AT_ERROR if they try to read the key.

```
AT+MAPPKEY=?
>>>AT_ERROR
```

Users can receive AT_PARAM_ERROR if the parameter has less than 16 bytes.

```
AT+APPKEY=112233445566778899
>>>AT_PARAM_ERROR
```

Users can receive AT_BUSY_ERROR if the command is used during LoRa transmission*. The user will have to wait until the transmission is done.

8.7.5.4 AT+MJOIN: Join Multicast Network

This command sends a multicast join request to the network. This command will send the Join Request and wait for RX1DL milliseconds for the RX1 window to listen for a 'Multicast Join Accept' packet from the network server. The join request will use datarate set by AT+DR (default DR0).

The client module takes one attempt to join a multicast group and then go back to sleep. An example of how to use this command can be found in section 10.7.

Note: Users can send a multicast join request anytime they want, even if the device is already joined, to update the session keys.

```
AT+MJOIN?
>>>Join Multicast Network
>>>OK
```

// Case: Successfully joined

```
AT+MJOIN
>>>OK
>>>txDone
>>>rxDone
AT+MJS=?
>>>JOINED
>>>OK
```

// Case: Failed to join

```
AT+MJOIN
>>>OK
>>>txDone
>>>rx1Timeout
>>>rx2Timeout
AT+MJS=?
>>>NOT JOINED
>>>OK
```

Users can receive AT_BUSY_ERROR if the command is used during a LoRa transmission*, AT_INVALID_MODE if the MODE is not LORAWAN and AT_PARAM_ERROR if the class isn't C.

*LoRa transmission = Join procedure, single packet uplink transmission or continuous transmission

```
AT+CLASS=A
>>>OK
AT+MJOIN
>>>AT_PARAM_ERROR
```

Users can receive AT_NO_NETWORK_JOINED if the command is used before the device joined a unicast network.

```
AT+NJS=?
>>>NOT JOINED
>>>OK
AT+MJOIN
>>>AT_NO_NETWORK_JOINED
```

8.7.5.5 AT+MJS: Multicast Join Status

This command provides users the join status of the multicast. It is a read-only command. The response can be either JOINED or NOT JOINED. The command provides the last multicast join request status.

```
AT+MJS?
>>>Get Multicast Join Status
>>>OK
AT+MJOIN
>>>OK
>>>txDone
>>>rxDone
AT+MJS=?
>>>JOINED
>>>OK
```

8.7.5.6 AT+MDEVADR: Device Address

This command allows users to access the multicast device address. Multicast join will provide the MDEVADR. It is a read-only command.

```
AT+MDEVADR?
>>>Get Multicast Device Address
>>>OK
AT+DEVADR=?
>>>11:22:33:AA
>>>OK
```

8.7.5.7 AT+MTIMER: Multicast Keep-Alive Timer

This command provides users to access the keep-alive timer. This command takes hours as input. The default value is 24 hours. Setting it zero will turn off the timer.

```
AT+MTIMER?
>>>Config Multicast Keep-Alive Timer
>>>OK
AT+MTIMER=?
>>>24
>>>OK
AT+MTIMER=1
>>>OK
```

8.7.6 LoRa Only Commands

This section provides a set of commands to use the LoRa physical layer to send and receive radio packets, bypassing the LoRaWAN protocol.

8.7.6.1 AT+MODE: Switch Between LoRa and LoRaWAN

The command allows users to switch between the LoRa and LoRaWAN mode. The default mode is LoRaWAN. Changing the mode to LoRa will allow users to use LoRa-only specific commands.

```
AT+MODE?
>>>Switch between LoRa & LoRaWAN
>>>OK
AT+MODE=?
>>>LORAWAN                    // LoRaWAN Enabled (Default)
>>>OK
AT+MODE=LORA
>>>OK
```

Users can receive `AT_PARAM_ERROR` if the input parameter is neither `LORA` nor `LORAWAN` and `AT_BUSY_ERROR` if the command is used during a LoRa transmission*.

8.7.6.2 AT+RFCFG: LoRa Packet Configuration

This command allows users to change the parameters for LoRa packet transmission and reception. This command doesn't depend on `AT+MODE` configuration.

Format: `AT+RFCFG=<Freq Hz>,<SF>,<BW>,<PWR>,<Rx Preamble>,<Tx Preamble>`

Table 8-2: Parameter ranges for LoRa only commands

Parameters	HXC400	HXC900
Frequency	137 – 525 MHz	860 – 1020 MHz
Spreading Factor	6 – 12	
Bandwidth	125, 250, 500 KHz	
Power	-1 to 20 dBm	
Rx and Tx Preamble	6 to 65535	

```
AT+RFCFG?
>>>Config LoRa Only Parameters
>>>OK
AT+RFCFG=?
>>> Freq=902300000Hz // HXC900
>>> SF=10
>>> Bandwidth=125kHz
>>> Power=20dBm
>>> RxPreamble=10
>>> TxPreamble=10
>>> OK
```

*LoRa transmission = Join procedure, single packet transmission or continuous transmission

```
AT+RFCFG=903900000,10,250,17,10,10 // HXC900
>>>OK
```

Users can receive AT_PARAM_ERROR if the command format is not right or parameters are out of range.

```
AT+RFCFG=903900000,10,250,17,10,10
>>>AT_PARAM_ERROR     // HXC400
```

8.7.6.3 AT+TX: Transmit LoRa Packet

This command allows users to send LoRa only packet (bypassing LoRaWAN) using the parameters set by AT+RFCFG. The command takes ASCII characters. The mode needs to be set to LORA (AT+MODE) before using this command. Format: AT+TX=<ASCII Bytes>

```
AT+TX?
>>>Send LoRa Uplink
>>>OK
AT+TX=hello
>>> OK
>>> txDone
```

Users can receive AT_BUSY_ERROR if the command is used during a LoRa transmission.

```
AT+TX=You are not going anywhere!
>>> AT_BUSY_ERROR
```

Users can receive AT_INVALID_MODE if the MODE is not LORA.

```
AT+TX=What mode are you?
>>>AT_INVALID_MODE
```

8.7.6.4 AT+RX: Receive LoRa Packet

This command allows users to receive LoRa packet (bypassing LoRaWAN) using the parameters set by AT+RFCFG. Turning on the reception will let users listen for LoRa packets continuously until they turn it off. The mode needs to be set to LORA (AT+MODE) before using this command. Format: AT+RX=<ON(1)/OFF(0)>

```
AT+RX?
>>>Receive LoRa Downlink
>>>OK
AT+RX=1      // Turn on the reception.
>>>OK
AT+RX=0      // Turn off the reception.
>>>OK
```

The received data will be printed using UART.

```
>>>rxDone
>>>rxdata:<Hex Bytes>
```

Users can receive AT_BUSY_ERROR if the command is used during a LoRa transmission.

```
AT+RX=1
>>> AT_BUSY_ERROR
```

Users can receive AT_INVALID_MODE if the MODE is not LORA.

```
AT+RX=1
>>>AT_INVALID_MODE
```

8.7.6.5 AT+TXCW: Continuous Transmit

This command allows the user to set the radio to output a fundamental frequency (unmodulated). It is mainly used for ETSI and FCC certification. It will use AT+RFCFG command's <Freq>, <SF>, <BW>, and <PWR> parameters. This command **doesn't** depend on AT+MODE configuration.

Format: AT+TXCW=<timeout in seconds>

```
AT+TXCW?
>>>Turn ON Continuous Tx
>>>OK
AT+TXCW=60
>>>OK
```

Users can receive AT_BUSY_ERROR if the command is used during a LoRa transmission.

```
AT+TXCW=10
>>> AT_BUSY_ERROR
```

*LoRa transmission = Join procedure, single packet transmission or continuous transmission

8.7.7 Diagnostic Commands

This section provides a set of commands to provide diagnostic information on the RF signal quality and battery level.

8.7.7.1 AT+BAT: Get Battery Level

This command allows users to get the battery level of the client. If the network server requests for device status (DevStatusReq) this battery level, along with the last received packet's SNR, will be incorporated in the 'DevStatusAns'.

Battery	Description
0	The end-device is connected to an external power source
1 to 254	Battery level, 1 being at a minimum and 254 is maximum.
255	The end-device was not able to measure the battery level.

```
AT+BAT?
>>>Get Battery Level
>>>OK
AT+BAT=?
>>>254
```

8.7.7.2 AT+RSSI: RSSI of Last Received Packet

This command allows users to read the RSSI (in dBm) of the last received downlink packet. It is a read-only command.

```
AT+RSSI?
>>>Get RSSI of Last Downlink
>>>OK
AT+RSSI=?
>>>21
AT+RSSI=0
>>>AT_ERROR
```

8.7.7.3 AT+SNR: Signal to Noise Ratio

This command allows the user to read the SNR (in dBm) of the last received downlink packet. It is a read-only command.

```
AT+SNR?
>>>Get SNR of Last Downlink
>>>OK
AT+SNR=?
>>>21
AT+SNR=0
>>>AT_ERROR
```

8.7.7.4 AT+VER: Version of the Firmware

This command allows the user to get the version of the HXC client module firmware. The *Get* command will provide the version number. It is a read-only command.

```
AT+VER?
>>>Get FW Version
>>>OK

AT+VER=?
>>>1.18
>>>OK
```

9 Appendix I: Interfacing HXC Client using Expansion Board

Our Embedded API for HXC Client helps users to integrate HXC Client into their End node product. But for debugging purposes, users might want to execute AT commands using a terminal software. To do that, users will need a USB-to-Serial interface to communicate with an HXC Client Module. An example of such a connection is described in 6.2. Our HXC Client Expansion Board can be used with either Arduino or STM32 Nucleo Development Boards. We can turn the Development boards into a USB-to-Serial converter.

Figure 9-1: HXC Client Expansion Board

The following steps describe how we can use STM32 Nucleo-L053R8 as USB-to-Serial device:

- Go to our HXC Client Expansion Board page. Open the user guide found at the bottom of that article.
- Follow 'Chapter 4: Quick Start' until step 11. Step 11 asks you to import the Embedded API. In our case, as we merely want to use our dev board as a USB-to-Serial device. So go to this repository instead.
- Follow the rest of the steps from that guide (except steps 13, 14, 18 and 19).

9.1 Interfacing using TeraTerm

- Open your device manager (`Control Panel->Device Manager`) to note down the COM port. Depending on what driver is installed, the Nucleo Board can show up as '`STMicroelectronics STLink Virtual COM Port`' or simply as '`USB Serial Interface`'.

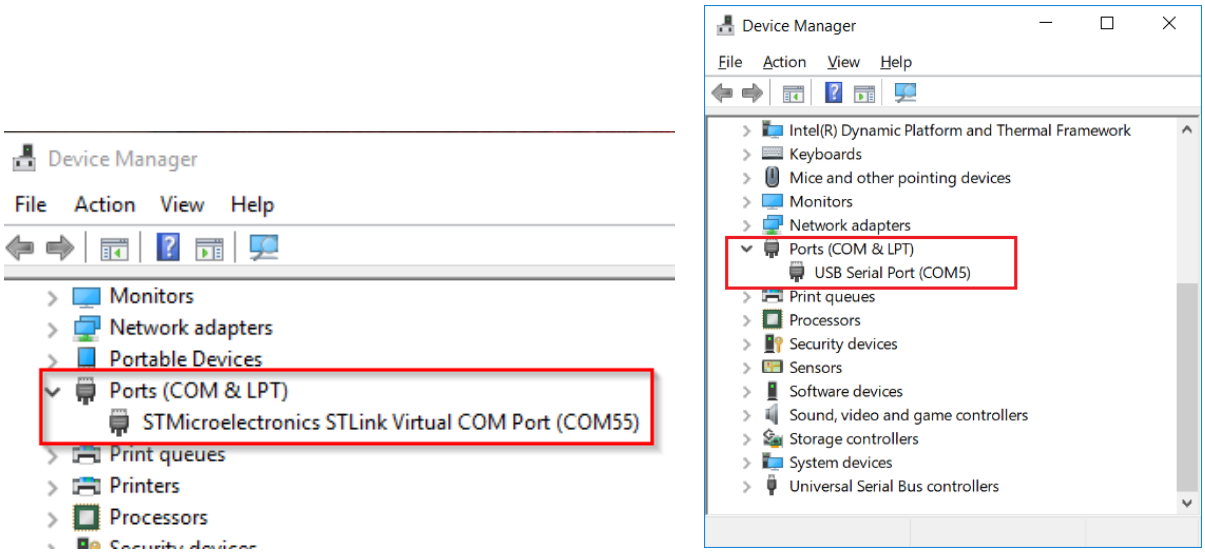


Figure 9-2: Nucleo board showed up as COM port

- Download TeraTerm. Once it is installed open it. Select appropriate COM port ('COM5' in our case) from 'Serial Port' drop-down menu.

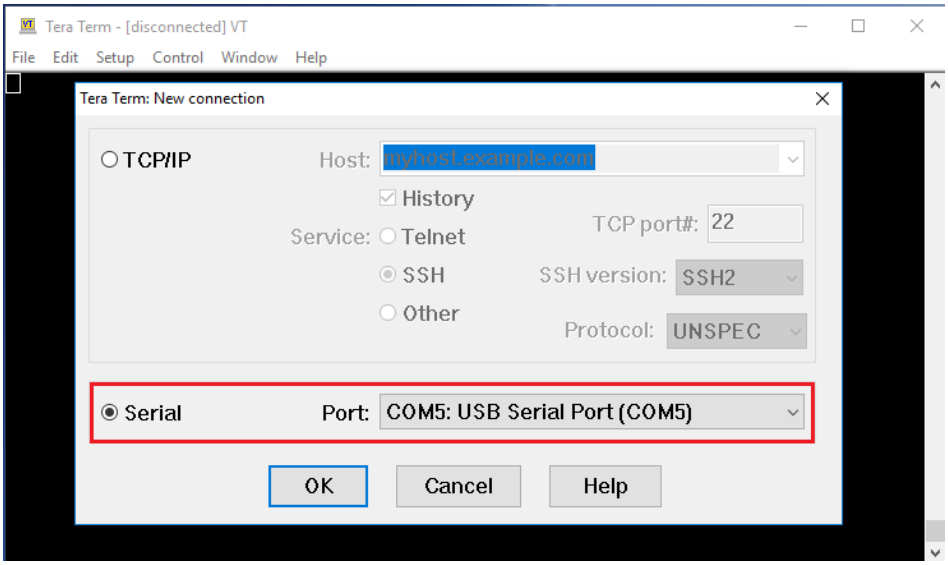


Figure 9-3: Tera Term connection window

- Go to `Setup->Terminal…` and make changes according to the figure below.
  Note: All AT commands need to end with <CR><LF>. Don't forget to enable 'Local Echo'.

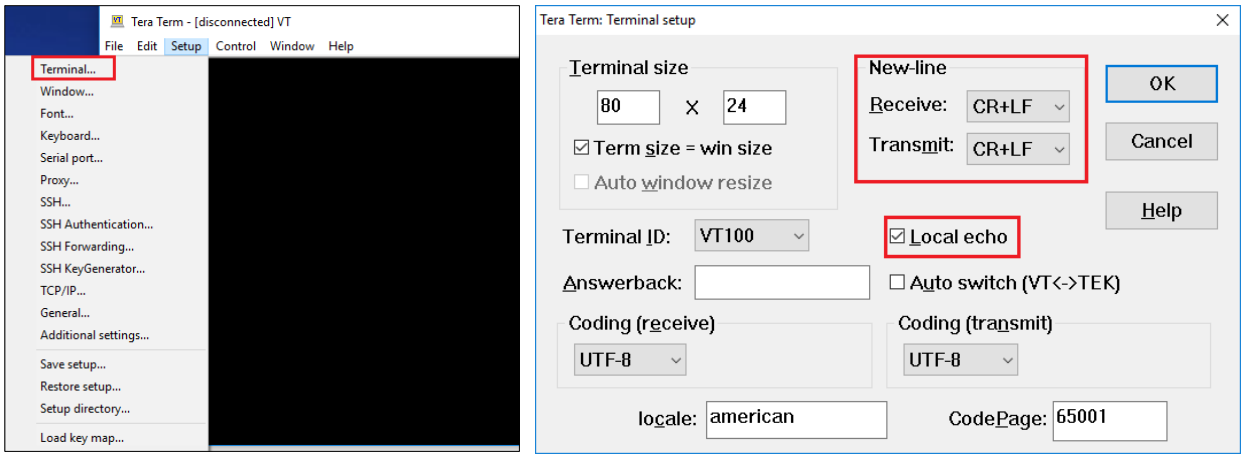


Figure 9-4: Tera Term terminal setup

- Go to `Setup->Serial port...` and change the serial port configuration.

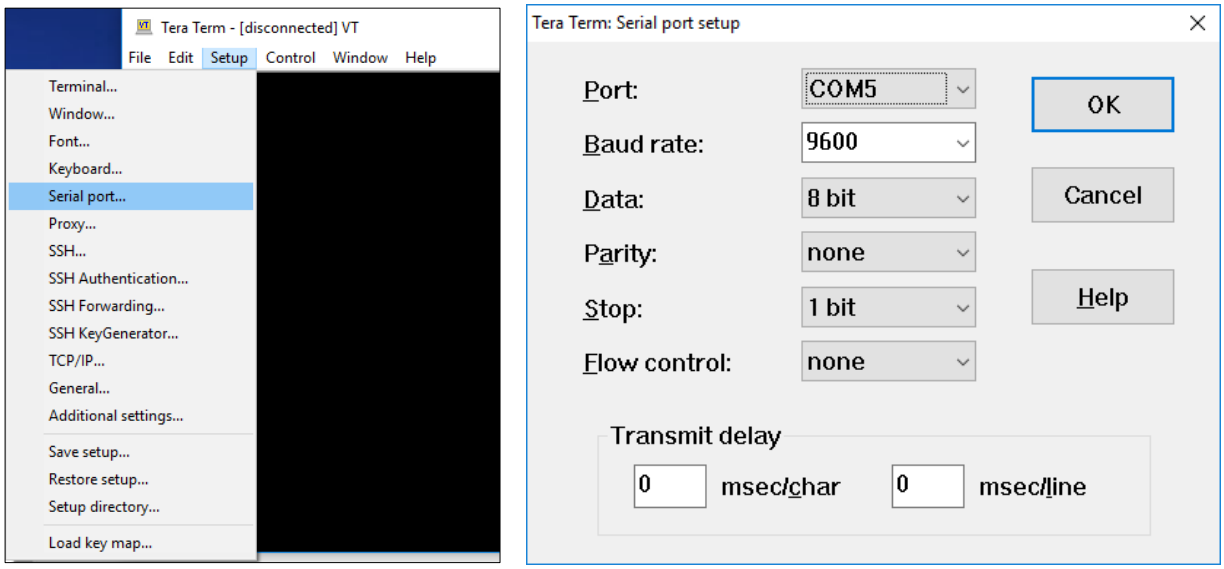


Figure 9-5: Tera Term Serial setup

- You can now type any AT commands and the terminal window will show responses.

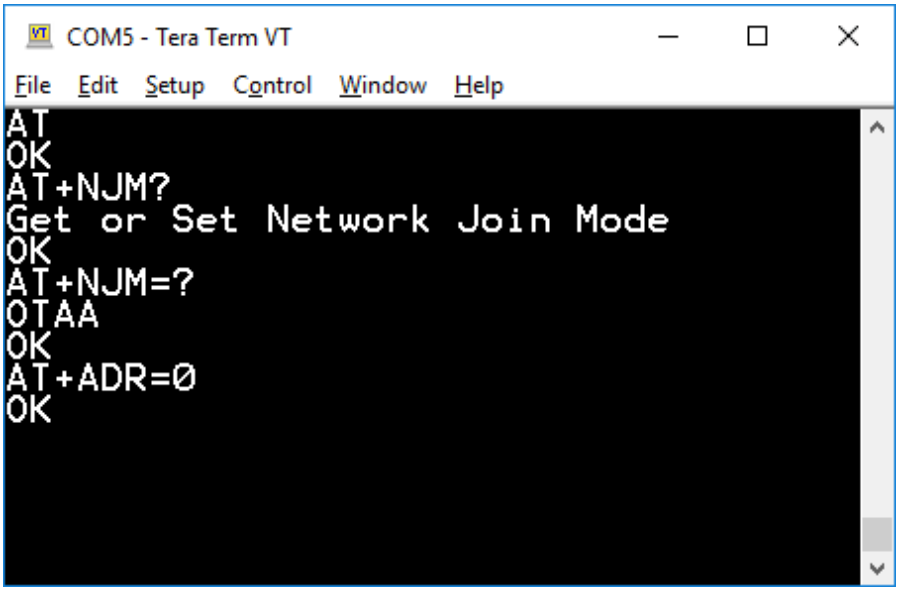


Figure 9-6: Executing AT commands

10 Appendix II: List of AT Commands for Operations

The following list shows the bare minimum AT commands to send a packet using the OTAA join mode. HXC400 and HXC900 commands are almost the same except the channel configuration commands.

To facilitate the example, all <CR><LF> is intentionally omitted. All commands will have to end with <CR><LF>.

10.1 HXC400 with Class A and OTAA

```
AT+FD                                   // Factory Reset the module, to make sure the module
>>>HXC400 AT Interface                  // starts with default values.
>>>FW Ver: 1.18
>>>OK
AT+NJM=OTAA                             // Set join mode
>>>OK
AT+CLASS=A                              // Set class
>>>OK
AT+CH=?                                 // RegionCN470 default channel configuration
>>>0,471500000,5,0
>>>1,471700000,5,0
>>>2,471900000,5,0
>>>3,472100000,5,0
>>>4,472300000,5,0
>>>5,472500000,5,0
>>>6,472700000,5,0
>>>7,472900000,5,0
>>>OK
AT+DEVEUI=be7a00000000042c              // Set Device EUI
>>>OK
AT+APPEUI=be7c000000000178              // Set Application EUI
>>>OK
AT+APPKEY=3e03deb24AD1053be54ca996c334408a
>>>OK
AT+JOIN                                 // Sending a Join Request to a Network Server
>>>OK
>>>txDone
>>>rxDone
>>>JOINED
AT+NJS=?                                // Check join status
>>>JOINED
>>>OK
AT+SENDB=0,12:CA73F00D                  // Sending an uplink with payload: CA73F00D (hex)
>>>OK
>>>txDone
>>>rx1Timeout                           // No downlink from NS
>>>rx2Timeout
```

10.2 HXC900 with Class A and OTAA

```
AT+FD                                   // Factory Reset the module, to make sure the module
>>>HXC900 AT Interface                  // starts with default values.
>>>FW Ver: 1.18
>>>OK
AT+NJM=OTAA                             // Set join mode
>>>OK
AT+CLASS=A                              // Set Class
>>>OK
AT+CH=?                                 // RegionUS915 Default channels configurations
>>>0,902300000,3,0
>>>1,902500000,3,0
>>>2,902700000,3,0
>>>3,902900000,3,0
>>>4,903100000,3,0
>>>5,903300000,3,0
>>>6,903500000,3,0
>>>7,903700000,3,0
>>>OK
AT+DEVEUI=be7a00000000042c
>>>OK
AT+APPEUI=be7c000000000178
>>>OK
AT+APPKEY=3e03deb24a835b3be54ca996c334408a
>>>OK
AT+JOIN
>>>OK
>>>txDone
>>>rxDone
>>>JOINED
AT+NJS=?                                // Check join status
>>>JOINED
>>>OK
AT+SENDB=0,21:CA7FA4CE!
>>>OK
>>>txDone
>>>rx1Timeout                           // No downlink from NS
>>>rx2Timeout
```

10.3 HXC900 with Class C and OTAA

```
AT+FD
>>>HXC900 AT Interface
>>>FW Ver: 1.18
>>>OK
AT+NJM=OTAA
>>>OK
AT+CLASS=C
>>>OK
AT+DEVEUI=be7a00000000042c
>>>OK
AT+APPEUI=be7c000000000178
>>>OK
AT+APPKEY=3e03deb24a835b3be54ca996c334408a
>>>OK
AT+JOIN
>>>OK
>>>txDone
>>>rxDone
>>>JOINED
AT+NJS=?                          // Check join status
>>>JOINED
>>>OK
AT+SENDB=0,21:C0FFEE
>>>OK
>>>txDone
>>>rx1Timeout                     // No downlink from NS
                                  // Class C always listens on receive window 2.
```

10.4 HXC400 with Class A and ABP

```
AT+FD                                   // Factory reset the module to make sure the module
>>>HXC400 AT Interface                  // starts with default values.
>>>FW Ver: 1.18
>>>OK
AT+NJM=ABP
>>>OK
AT+CLASS=A
>>>OK
AT+DEVEUI=BE7A00000000042C
>>>OK
AT+APPEUI=BE7C000000000178
>>>OK
AT+APPKEY=3E03DEB24A835B3BE54CA996C334408A
>>>OK
AT+DEVADR=C0010C0FFE
>>>OK
AT+APPSKEY=C87430800BADCA71233440000C001CA7
>>>OK
AT+NWKSKEY=DEA24D83FC3BE00CA996C3C001C0FFE8
>>>OK
AT+JOIN                                 // Although ABP doesn't require Join Request, users
>>>OK                                   // will have to execute the cmd nonetheless
>>>JOINED
AT+NJS=?
>>>JOINED
>>>OK
AT+SENDB=0,12:A3F4E1
>>>OK
>>>txDone
>>>rx1Timeout                           // No downlink from NS
>>>rx2Timeout
```

10.5 LoRa Transmission using HXC900

```
AT+FD                                   // Factory reset the module to make sure the module
>>>HXC900 AT Interface                  // starts with default values.
>>>FW Ver: 1.18
>>>OK
AT+MODE=LORA
>>>OK
AT+RFCFG=902300000,10,250,17,8,8
>>>OK
AT+TX=hello
>>>OK
>>>txDone
```

10.6 LoRa Reception using HXC900

```
AT+FD                                   // Factory reset the module to make sure the module
>>>HXC900 AT Interface                  // starts with default values.
>>>FW Ver: 1.18
>>>OK
AT+MODE=LORA
>>>OK
AT+RFCFG=902300000,10,250,17,8,8
>>>OK
AT+RX=1                                 // Turned on the receive channel
>>>OK
>>>rxDone
>>>rxdata:68656C6C6F                    // Hex bytes of 'hello'
```

10.7 Multicast with Class C and HXC900

```
AT+FD                                   // Factory reset the module to make sure the module
>>>HXC900 AT Interface                  // starts with default values.
>>>FW Ver: 1.18
>>>OK
AT+CLASS=C
>>>OK
AT+DEVEUI=be7a00000000042c
>>>OK
AT+APPEUI=be7c000000000178
>>>OK
AT+APPKEY=3e03deb24a835b3be54ca996c334408a
>>>OK
AT+JOIN
>>>OK
>>>txDone
>>>rxDone
>>>JOINED
AT+MGROUPID=BE7A00000042
>>>OK
AT+MGROUPKEY=CD7C0000010001781A2B
>>>OK
AT+MAPPKEY=3e03deb24a835b3be54ca996c334408a
>>>OK
AT+MJOIN                                // Don't forget to set the keys
>>>OK
>>>txDone
>>>rxDone
AT+MJS=?                                // Check multicast join status
>>>JOINED
>>>OK
AT+MTIMER=48                            // Update keep-alive timer to 48 hours
>>>OK
```

11 Appendix III: Regulatory Approval

This section outlines the regulatory information for the HXC Client Module 900Mhz (Model: HXC900) for the following countries:

- United States
- Canada

11.1 United States

The HXC900 module has received Federal Communications Commission (FCC) CFR47 Telecommunications, Part 15 Subpart C "Intentional Radiators" modular approval in accordance with Part 15.212 Modular Transmitter approval. Modular approval allows the end user to integrate the HXC900 module into a finished product without obtaining subsequent and separate FCC approvals for intentional radiation, provided no changes or modifications are made to the module circuitry. Changes or modifications could void the user's authority to operate the equipment. The end user must comply with all of the instructions provided by the Grantee, which indicate installation and/or operating conditions necessary for compliance.

The finished product is required to comply with all applicable FCC equipment authorizations regulations, requirements, and equipment function not associated with the transmitter module portion. For example, compliance must be demonstrated to regulations for other transmitter components within the host product; to requirements for unintentional radiators (Part 15 Subpart B "Unintentional Radiators"), such as digital devices, computer peripherals, radio receivers, etc.; and to additional authorization requirements for the non-transmitter functions on the transmitter module (i.e., Verification, or Declaration of Conformity) (e.g., transmitter modules may also contain digital logic functions) as appropriate.

11.1.1 Labeling and User Information Requirements

The HXC900 module has been labeled with its own FCC ID number, and if the FCC ID is not visible when the module is installed inside another device, then the outside of the finished product into which the module is installed must also display a label referring to the enclosed module. This exterior label can use wording as follows:

> Contains Transmitter Module FCC ID: 2ANQY-HXC900
> Contains Transmitter Module IC: 23185-HXC900
> Or
> Contains FCC ID: 2ANQY-HXC900
> Contains IC: 23185-HXC900
>
> This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

A user's manual for the finished product should include the following statement:

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Additional information on labeling and user information requirements for Part 15 devices can be found in KDB Publication 784748 available at the FCC Office of Engineering and Technology (OET) Laboratory Division Knowledge Database (KDB) https://apps.fcc.gov/oetcf/kdb/index.cfm.

11.1.2 RF Exposure

All transmitters regulated by FCC must comply with RF exposure requirements. KDB 447498 General RF Exposure Guidance provides guidance in determining whether proposed or existing transmitting facilities, operations or devices comply with limits for human exposure to Radio Frequency (RF) fields adopted by the Federal Communications Commission (FCC).

From the HXC900 FCC Grant: Output power listed is conducted. This grant is valid only when the module is sold to OEM integrators and must be installed by the OEM or OEM integrators. This transmitter is restricted for use with the specific antenna(s) tested in this application for Certification and must not be co-located or operating in conjunction with any other antenna or transmitters within a host device, except in accordance with FCC multi-transmitter product procedures.

11.1.3 Approved External Antenna Types

To maintain modular approval in the United States, only the antenna types that have been tested shall be used. It is permissible to use different antenna manufacturer provided the same antenna type and antenna gain (equal to or less than) is used.

Testing of the HXC900 module was performed with the antenna types listed in section 3.1.

11.1.4 Helpful Websites

Federal Communications Commission (FCC): http://www.fcc.gov

FCC Office of Engineering and Technology (OET) Laboratory Division Knowledge Database (KDB): https://apps.fcc.gov/oetcf/kdb/index.cfm

11.2 Canada

The HXC900 module has been certified for use in Canada under Industry Canada (IC) Radio Standards Specification (RSS) RSS-210 and RSS-Gen. Modular approval permits the installation of a module in a host device without the need to recertify the device

11.2.1 Labeling and User Information Requirements

Labeling Requirements for the Host Device (from Section 3.2.1, RSS-Gen, Issue 3, December 2010): The host device shall be properly labeled to identify the module within the host device.

The Industry Canada certification label of a module shall be clearly visible at all times when installed in the host device, otherwise the host device must be labeled to display the Industry Canada certification number of the module, preceded by the words "Contains transmitter module", or the word "Contains", or similar wording expressing the same meaning, as follows:

> Contains Transmitter Module FCC ID: 2ANQY-HXC900
> Contains Transmitter Module IC: 23185-HXC900
> Or
> Contains FCC ID: 2ANQY-HXC900
> Contains IC: 23185-HXC900

User Manual Notice for License-Exempt Radio Apparatus (from Section 7.1.3 RSS-Gen, Issue 3, December 2010): User manuals for license-exempt radio apparatus shall contain the following or equivalent notice in a conspicuous location in the user manual or alternatively on the device or both:

> This device complies with Industry Canada license-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.
>
> Le présent appareil est conforme aux CNR d'Indu-strie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes: (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

Transmitter Antenna (from Section 7.1.2 RSS-Gen, Issue 3, December 2010): User manuals for transmitters shall display the following notice in a conspicuous location:

> Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that necessary for successful communication.
>
> Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensité nécessaire à l'établissement d'une ommunication satisfaisante

The above notice may be affixed to the device instead of displayed in the user manual. User manuals for transmitters equipped with detachable antennas shall also contain the following notice in a conspicuous location:

> This radio transmitter (identify the device by certification number, or model number if Category II) has been approved by Industry Canada to operate with the antenna types listed below with the maximum permissible gain and required antenna impedance for each antenna type indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.
>
> Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensité nécessaire à l'établissement d'une communication satisfaisante.

Immediately following the above notice, the manufacturer shall provide a list of all antenna types approved for use with the transmitter, indicating the maximum permissible antenna gain (in dB) and required impedance for each.

11.2.2 RF Exposure

All transmitters regulated by IC must comply with RF exposure requirements listed in RSS-102 - Radio Frequency (RF) Exposure Compliance of Radiocommunication Apparatus (All Frequency Bands). Currently, this device is approved for use for when 20 cm can be maintained between the antenna and users.

Specific Absorption Rate (SAR) evaluation is required if the separation distance between the user and/or bystander and the antenna and/or radiating element of the device is less than or equal to 20 cm. Exceptions are listed in RSS-102. Note that integration < 20 cm will require further certification with IC such as a Multiple listing and Class IV Permissive Change application.

11.2.3 Approved External Antenna Types

Transmitter Antenna (from Section 7.1.2 RSS-Gen, Issue 3, December 2010):
The HXC900 module can only be sold or operated with antennas with which it was approved. The transmitter may be approved with multiple antenna types. An antenna type comprises antennas having similar in-band and out-of-band radiation patterns. Testing shall be performed using the highest gain antenna of each combination of transmitter and antenna type for which approval is being sought, with the transmitter output power set at the maximum level. Any antenna of the same type having equal or lesser gain as an antenna that had been successfully tested with the transmitter will also be considered approved with the transmitter and may be used and marketed with the transmitter.

When a measurement at the antenna connector is used to determine RF output power, the effective gain of the device's antenna shall be stated, based on measurement or on data from the antenna manufacturer. For transmitters of output power greater than 10 milliwatts, the total antenna gain shall be added to the measured RF output power to demonstrate compliance with the specified radiated power limits.

Testing of the RN2903 module was performed with the antenna types listed in section 3-1.

11.2.4 Helpful Websites

Industry Canada: http://www.ic.gc.ca/

12 Appendix IV: Document Information

12.1 Version History

Version	Date	Author	Description
V1.00	07/07/17	Fahad Mirza	• Initial Release.
V1.01	08/06/17	Fahad Mirza	• Dimension and Connection diagram added.
V1.02	08/10/17	Fahad Mirza	• List of error codes added.
V1.03	08/14/17	Fahad Mirza	• Updated AT command examples.
V1.04	08/15/17	Fahad Mirza	• Appendix I, II, III added.
V1.05	08/16/17	Fahad Mirza	• Updated AT commands. • Updated Table of Contents.
V1.06	08/22/17	Fahad Mirza	• Added +RESET, +FD, +RX1FR, +TXCW. • Updated region-specific parameter table. • Added WC message section. • Fixed several AT command ranges.
V1.07	10/12/17	Fahad Mirza Nik Kitson	• RESET pin info updated. • DEVEUI error response updated. • Physical Characteristics added.
V1.08	11/02/17	Fahad Mirza	• Pinout diagram added. • HXC900 parameter table added.
V1.09	01/17/18	Fahad Mirza Nik Kitson	• US915 channel configuration added. • Updated appendix examples. • AT+BAT added. • AT+PAYLOAD_SIZE_ERROR added. • FCC and IC statements added. • Antenna description added. • List of figures and table added.
V1.10	4/13/18	Fahad Mirza	• AT+VER updated. • Added payload size corresponding to data rates. • AT+RESET and AT+FD response updated. • Power consumption table updated. • Added new and updated previous examples. • Added AT+RECVB command. • AT+SEND, +DR, RX2WND, +JOIN updated. • Updated downlink packet information.
V1.11	08/09/18	Fahad Mirza	• Added HXC900 Class A example. • Added bit/s for datarate/spreading factor. • Updated with Embedded API link. • Removed Client Manager support.

			• Updated Appendix I with HXC Client Expansion board. • Appendix II AT commands is linked to the appropriate section of the document.
V2.00	11/05/18	Fahad Mirza	• Updated the default channel list of HXC400. • Added AT_NO_CHANNEL_ERROR status info. • Added LoRa only example. • Added multicast commands. • Added multicast example.
V2.01	01/09/19	Fahad Mirza	• Removed ST-Link Connection diagram. • Updated pin layout with NC. • Added module weight.

12.2 List of Abbreviations

Acronym	Definition
LoRa®	Long range radio modulation scheme
LoRaWAN™	LoRa® wide-area network protocol
RF	Radio frequency
RSSI	Received signal strength indicator
SNR	Signal to noise ratio
OTAA	Over-the-air Activation
ABP	Activation by personalization
ETSI	European telecommunications standards institute
FCC	Federal Communications Commission
AES	Advanced Encryption Standard
NS	Network Server