LoRaWAN® AT-команды для STM32CubeWL

Вступление

В этой заметке по применению объясняется, как взаимодействовать с LoRaWAN® для управления беспроводным каналом LoRa® с помощью AT-команд.

В этом документе перечислен набор AT-команд на платах NUCLEO_WL55JC STM32WL Nucleo (коды заказа NUCLEO - WL55JC1 для диапазона высоких частот и NUCLEO-WL55JC2 для диапазона низких частот).

Прошивка пакета MCU STM32CubeWL основана на драйверах STM32Cube HAL.

1 Общая информация

STM32CubeWL работает на микроконтроллерах серии STM32WL на базе процессора Arm® Cortex®-M.

Примечание. Arm является зарегистрированным товарным знаком Arm Limited (или ее дочерних компаний) в США и / или в других странах.

Таблица 1. Аббревиатуры

Акроним	Определение
ABP	Активация путем персонализации
LoRa	Радиотехника дальнего действия
LoRaWAN	Глобальная сеть LoRa
OTAA	Активация по воздуху
RSSI	Индикатор уровня принимаемого сигнала
SNR	Соотношение сигнал / шум

Справочные документы

[1] LoRaWAN 1.0.3 Specification by LoRa Alliance® Specification Protocol - январь 2018 г.

[2] Примечание по приложению Как создать приложение LoRa с STM32CubeWL (AN5406)

[3] Руководство пользователя Описание STM32WL HAL и низкоуровневых драйверов (UM2642)

2 Обзор

Платы NUCLEO-WL55JC, STM32WL Nucleo содержат набор AT-команд для тестирования LoRa RF и обмена данными LoRaWAN.

В этом примечании к приложению подробно описывается интерфейс, определение AT-команд, некоторые варианты использования и описание встроенного программного обеспечения. Полное описание приложения LoRa, созданного с помощью STM32CubeWL, см. В документе [2]

3 AT-команды

Команды AT используются для управления модулем LoRa и для отправки данных (см. Документ [1] для получения более подробной информации).

AT-команды отправляются через периферийное устройство UART.

В демонстрации ниже хост (обычно хост Windows®) может быть подключен к модулю с помощью ST-LINK. Затем можно использовать UART через ST-LINK

со стандартным программным обеспечением Windows, таким как TeraTerm или PuTTY) со следующими параметрами:

- Скорость передачи: 9600
- Данные: 8 бит
- Четность: нет
- Стоп: 1 бит
- Контроль потока: нет

Вот типичная конфигурация Tera Term:

Figure 1. Tera Term serial port set up

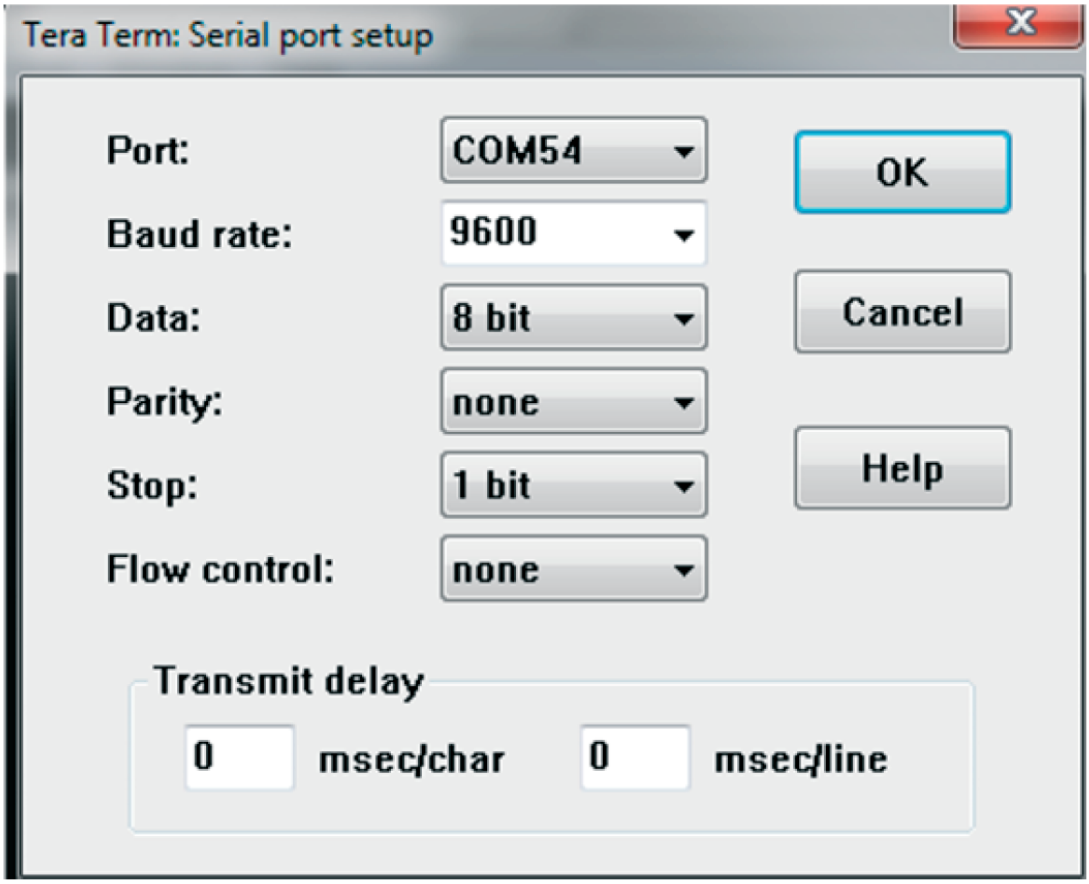
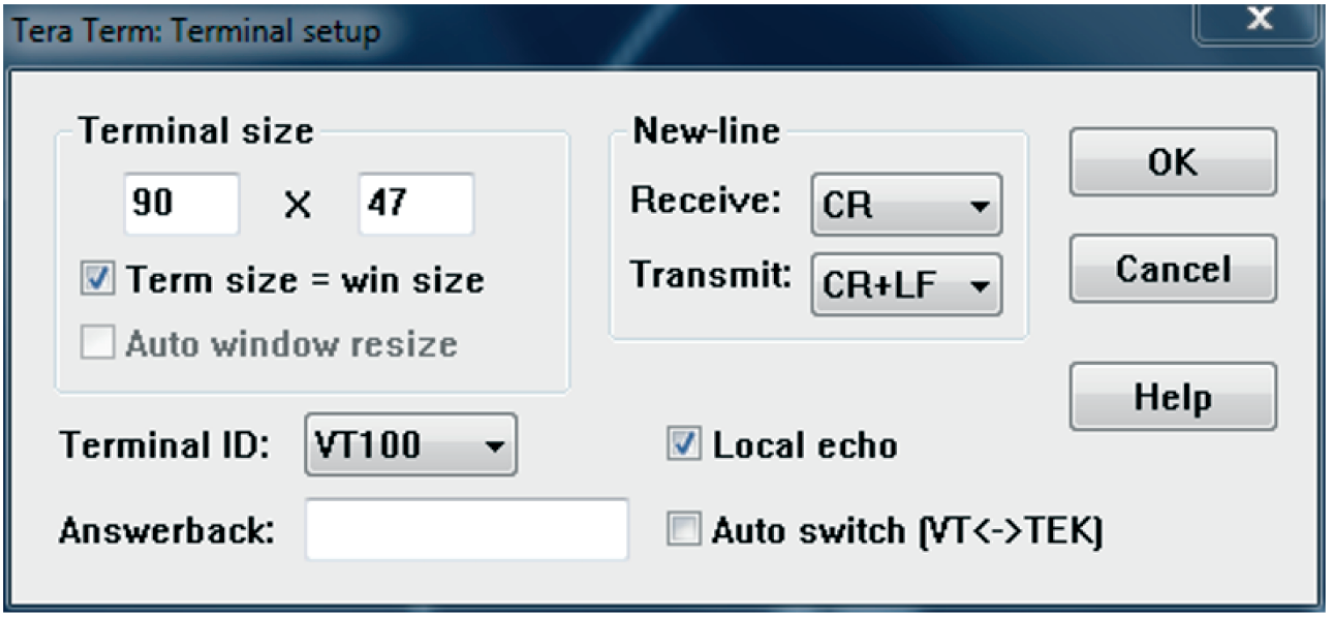


Figure 2. Tera Term terminal setup



Все команды имеют форму AT + XXX, где XXX обозначает команду. Доступны следующие варианты поведения команд:

• AT + XXX? предоставляет краткую справку по данной команде (например, AT + DEUI?).

• AT + XXX используется для запуска команды (например, AT + JOIN).

• AT + XXX =? используется для получения значения заданной команды (например, AT + CFS =?).

• AT + XXX = <значение> используется для предоставления значения команде (например, AT + SEND = 2: Hello).

Вывод команд осуществляется через UART. Формат вывода обычно:

<value><CR><LF>
<CR><LF><Status<CR><LF>

Учитывая:

• <value> <CR> <LF> возвращается, когда справка AT + XXX? и получите AT + XXX =? команды выполняются.

• <CR> и <LF> обозначают возврат каретки и перевод строки.

• Если значение не возвращается, то <value> <CR> <LF> вообще не возвращается.

• Каждая команда, кроме ATZ (сброс MCU), возвращает строку состояния, которой предшествует и за ней следует <CR> <LF>. Возможный статус:

- OK: команда выполняется правильно, без ошибок.
- AT_ERROR: общая ошибка
- AT_PARAM_ERROR: неверный параметр команды.
- AT_BUSY_ERROR: сеть LoRa занята, поэтому команда не может быть выполнена.
- AT_TEST_PARAM_OVERFLOW: параметр слишком длинный.
- AT_NO_NETWORK_JOINED: сеть LoRa не подключена.
- AT_RX_ERROR: обнаружение ошибки при приеме команды

В следующих разделах описывается каждая команда, включая несколько примеров. Каждая команда, перед которой стоит знак #, передается модулю хостом. Затем распечатывается возврат модуля.

AT_ERROR возвращается, если команда не распознается.

3.1 AT_RX_ERROR

В случае AT_RX_ERROR команда повреждена при получении в AT_Slave. Следовательно, команда не выполняется.

Однако в случае длинных команд некоторые ложные символы все еще могут находиться в очереди, готовые к обработке как команды. Таким образом, в случае, если пользователь получает AT_RX_ERROR, он должен сначала отправить <CR> <LF> для очистки очереди, а затем отправить обратно ту же команду, чтобы она была обработана.

Пример

AT + APPKEY = 2b: 7e: 15: 16: 28: ae: d2: a6: ab: f7: 15: 88: 09: cf: 4f: 3c <CR> <LF>
<CR> <LF> AT_RX_ERROR <CR> <LF> / * обнаружена ошибка RX * /
<CR> <LF> AT_ERROR <CR> <LF> / * после команды AT_Slave обработал «что-то», * /
/ * что является не командой - это может привести к ошибке * /

<CR> <LF> / * новая строка для очистки * /
<CR> <LF> AT_ERROR <CR> <LF> / * очистка может привести к ошибке * /
/ * теперь можно повторно отправить команду * /
AT + APPKEY = 2b: 7e: 15: 16: 28: ae: d2: a6: ab: f7: 15: 88: 09: cf: 4f: 3c <CR> <LF>

3.2 Обзор AT-команд

Таблица 2. AT-команды

Команда	Параметры	Описание
Общие команды		
AT	Нет	Проверяет, доступен ли интерфейс.
AT	[?]	Справка по всем поддерживаемым командам.
ATZ	Нет	Сброс настроек
AT+VL	[=verb_lvl], где verb_lvl = [0:3]	Устанавливает / получает уровень подробности.
AT+LTIME	[=?]	Получает местное время в формате UTC.
Ключи, идентификаторы и команды управления EUI		
AT+APPEUI	[=01:02:03:04:05:06:07:08]	Устанавливает / получает приложение EUI.
AT+NWKKEY	[=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C]	Устанавливает / получает корневой ключ сети
AT+APPKEY	[=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C]	Устанавливает / получает корневой ключ приложения.
AT+APPSKEY	[=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C]	Устанавливает / получает ключ сеанса приложения.
AT+NWKSKEY	[=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C]	Устанавливает / получает ключ сетевой сессии.
AT+DADDR	[=01:02:0A:0B]	Устанавливает / получает адрес устройства.
AT+DEUI	[=01:23:45:67:89:AB:CD:EF]	Устанавливает / получает уникальный идентификатор модуля.
AT+NWKID	[=127]	Устанавливает / получает идентификатор сети.
LoRa команды приема и передачи данных		
AT+JOIN	[=mode] где режим = 0 (ABP) или режим = 1 (OTAA)	Присоединяется к сети.
AT+LINKC	-	Запрос команды MAC для проверки совмещенной ссылки на следующую ссылку
AT+SEND	[=port_nb:confirmedmode:data] где подтвержден режим = 0 или 1.	Отправляет пакеты в сеть.
Команды управления сетью LoRa		
AT+VER	[=?]	Получает версию LoRaWAN.
AT+ADR	[=adr_enable] где adr_enable = 0 or 1	Устанавливает / получает функцию адаптивной скорости передачи данных.
AT+DR	[=datarate] где datarate = [0:7]	Устанавливает / получает скорость передачи данных.
AT+BAND	[=region] где region = [0:9]	Устанавливает / получает активную область
AT+CLASS	[=class] где class = [A, B or C]	Устанавливает / получает класс LoRa.
AT+DCS	[=dutycycle] где dutycycle = 0 или 1	Устанавливает / получает настройки рабочего цикла.
AT+JN1DL	[=delay] где задержка в мс	Устанавливает / получает задержку соединения в окне Rx 1.
AT+JN2DL		Устанавливает / получает задержку соединения в окне Rx 2.
AT+RX1DL		Устанавливает / получает задержку окна Rx 1.
AT+RX2DL	[=delay] где задержка в мс	Устанавливает / получает задержку окна Rx 2.
AT+RX2DR	[=datarate] где X = [0:7]	Устанавливает / получает скорость передачи данных окна Rx 2.

AT+RX2FQ	[=freq] где частота в Гц	Устанавливает / получает частоту окна Rx 2.
AT+TXP	[=txpow] где txpow = [0:7]	Устанавливает / получает мощность передачи.
AT+PGSLOT	[=periodicity]	Устанавливает / получает слот ping.
Команды радиотестирования		
AT+TTONE	Нет	Устанавливает тест радиочастотного тона.
AT+TRSSI		Устанавливает тональный тест RF RSSI.
AT+TCONF	[=freq:pow:bw:sf:cr:lna:pa:mod:paylen:freqdev:lowdropt:BT] [=868000000:14:125:12:4/5:0:0:1:255:0:0:0 for example	Устанавливает / получает конфигурацию теста LoRa RF.
AT+TTX	[=nb_packets_sent]	Устанавливает количество пакетов, отправляемых для теста PER RF Tx.
AT+TRX	[=nb_packets_received]	Устанавливает количество пакетов, которые должны быть получены для теста PER RF Rx.
AT+CERTIF	[=mode] где режим = 0 (ABP) или режим = 1 (OTAA)	Устанавливает модуль в сертификацию LoRaWAN с режимом соединения.
AT+TTH	[=<Fstart>, <Fstop>, <FDelta>,<PacketNb>]	Запускает тест скачкообразной перестройки RF Tx от Fstart до Fstop (в Гц или МГц), Fdelta в Гц
AT+TOFF	Нет	Останавливает радиочастотные тесты.
Информационная команда		
AT+BAT	Нет	Получает уровень заряда батареи.

3.3 Таблица событий

В таблице ниже подробно описаны события, которые приложение AT_Slave отправляет в качестве уведомления хост-модулю.

Таблица 3. Таблица событий

Event	Return value	Description
+EVT:JOINED	None	Уведомляет, что хост-модуль был присоединен к шлюзу с помощью OTAA.
+EVT:JOIN FAILED	None	Уведомляет, что хост-модуль не завершил транзакцию присоединения (ошибка ID / ключей, Tx не получен шлюзом, Rx не получен или не расшифрован). В этом случае необходимо отозвать AT + JOIN.
+EVT:	:<port>:<size>:<payload>	Уведомляет хост-модуль о том, что в окне RX был получен асинхронный фрейм с фреймом нисходящей линии связи.
+EVT:	RX_<slot>:<DR>:<RSSI>:<SNR>	Уведомляет хост-модуль о получении асинхронного кадра в окне RX с параметрами нисходящего канала.
+EVT:	RX_<slot>:<DR>:<RSSI>:<SNR>:<DMODM>:<GWN>	Уведомляет хост-модуль о том, что в окне RX был получен асинхронный кадр с расширенными параметрами нисходящего канала. Это событие заменяет предыдущее событие, когда был выполнен хотя бы один запрос проверки ссылки (AT + LINKC).
+EVT:SEND_CONFIRMED	None	Уведомляет хост-модуль о том, что шлюз подтвердил фрейм Tx.

3.4 Общие команды

3.4.1 AT

Описание	Внимание используется для проверки правильности работы ссылки.
Синтаксис	AT<CR>
Аргументы	Нет
Ответ	Нет
Код результата	<CR><LF>OK<CR><LF>

Example:

/* Пример: проверьте правильность работы AT-ссылки */
AT<CR>
<CR>
OK<CR>

3.4.2 AT?

Описание	Предоставляет краткую справку по всем поддерживаемым командам.
Синтаксис	AT?<CR>
Аргументы	Нет
Ответ	Нет
Код результата	<CR><LF>OK<CR><LF>

Пример:

/ * Пример: получить краткую справку по ВСЕМ AT-командам * /
AT?<CR>
AT+<CMD>?
AT+<CMD> : Run <CMD>
AT+<CMD>=<value> : Set the value
AT+<CMD>=? : Get the value
<List of all commands help>
<CR>
OK<CR>

3.4.3 ATZ - MCU reset

Описание	Команда генерирует сброс NVIC: сбрасывает всю систему, включая радио и микропроцессор.
Синтаксис	ATZ<CR>
Аргументы	Нет
Ответ	Нет
Код результата	Нет (действие NVIC_Reset)

Пример:

/ * Пример: произвести сброс системы NVIC * /# ATZ<CR>
APP_VERSION: V1.1.0<CR>
MW_LORAWAN_VERSION: V2.3.0<CR>
MW_RADIO_VERSION: V1.1.0<CR>
DevEui: AA:BB:CC:DD:EE:FF:00:11<CR>

AppEui: 01:02:03:04:05:06:07:08<CR>
DevAddr: 12:34:56:78<CR>
ATtention command interface<CR>
AT? to list all available functions<CR>

Примечание. Отображаемые ключи по команде выше после ###### (DevEUI, AppEui и DevAddr) являются просто информативными, а не ответом на команду.

3.4.4 AT+VL - Уровень подробностей

Описание	Устанавливает / получает уровень подробности приложения.
Синтаксис	AT+VL=<verbose_leve><CR> AT+VL=?<CR>
Аргументы	<verbose_level>,по умолчанию 2 (VLEVEL_M) 0: VLEVEL_OFF 1: VLEVEL_L 2: VLEVEL_M 3: VLEVEL_H
Ответ	<verbose_level><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: установить уровень подробности * /
AT+VL=3<CR>
<CR>
OK<CR>
/ * Пример2: получить уровень подробности * /
AT+VL =?<CR>
3<CR>
<CR>
OK<CR>

3.4.5 AT + LTIME - Местное время в формате UTC.

Описание	Получает местное время в формате UTC.
Синтаксис	AT+LTIME=?<CR>
Аргументы	Нет
Ответ	<local time><CR><LF>
Код результата	<CR><LF>OK<CR><LF>

Пример:

/ * Пример: получить местное время в формате UTC * /
#AT+ LTIME =?<CR>
LTIME:02h14m52s on 01/01/1970<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.5 Управление ключами, идентификаторами и EUI

3.5.1 AT + APPEUI - идентификатор приложения

Описание	Устанавливает / получает приложение EUI.
Синтаксис	AT+APPEUI=<id><CR> AT+APPEUI=?<CR>
Аргументы	<id>, 8-байтовое значение, разделенное ":" (строка в шестнадцатеричном формате)
Ответ	<id><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/* Пример1: установить APP EUI */
AT+APPEUI=01:02:03:04:05:06:07:08<CR>
<CR>
OK<CR>
/* Пример2: получить APP EUI */
AT+APPEUI=?<CR>
01:02:03:04:05:06:07:08<CR>
<CR>
OK<CR>

3.5.2 AT + NWKKEY - сетевой корневой ключ

Описание	Устанавливает / получает корневой ключ сети. Этот ключ используется только в режиме OTAA.
Синтаксис	AT+NWKKEY=<key><CR> AT+NWKKEY=?<CR>
Аргументы	<id>, 4-байтовое значение, разделенное ":" (строка в шестнадцатеричном формате)
Ответ	<key><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/* Пример1: установить ключ NWK */
AT+NWKKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>
/* Пример 2: получить ключ NWK при #define KEY_EXTRACTABLE 1 */
AT+NWKKEY=?<CR>
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>

/* Пример 3: получить ключ NWK при #define KEY_EXTRACTABLE 0 */
AT+NWKKEY=?<CR>
<CR>
AT_ERROR<CR>

3.5.3 AT + APPKEY - корневой ключ приложения

Описание	Устанавливает / получает корневой ключ приложения. Этот ключ используется только в режиме OTAA.
Синтаксис	AT+APPKEY=<key><CR> AT+APPKEY=?<CR>
Аргументы	<key>, 16-байтовое значение, разделенное знаком «:» (строка в шестнадцатеричном формате)
Ответ	<key><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: установить ключ приложения * /
AT+APPKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>
/ * Пример2: получить ключ приложения, когда #define KEY_EXTRACTABLE 1 * /
AT+APPKEY=?<CR>
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>
/ * Пример 3: получить ключ приложения, когда #define KEY_EXTRACTABLE 0 * /
AT+APPKEY=?<CR>
<CR>
AT_ERROR<CR>

3.5.4 AT + APPSKEY - ключ сеанса приложения

Описание	Устанавливает / получает ключ сеанса приложения. Этот ключ используется только в режимах OTAA и APB. В режиме OTAA этот ключ заменяется в процессе создания корневым ключом приложения и информацией ответа JoinAccept.
Синтаксис	AT+APPSKEY=<key><CR> AT+APPSKEY=?<CR>
Аргументы	<key>, 16-байтовое значение, разделенное ":" (строка в шестнадцатеричном формате)
Ответ	<key><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Пример:

/ * Пример1: установить ключ сеанса приложения * /
AT+APPSKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>
/ * Пример 2: получить ключ сеанса приложения, когда #define KEY_EXTRACTABLE 1 * /
AT+APPSKEY=?<CR>
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>
/ * Пример 3: получить ключ сеанса приложения, когда #define KEY_EXTRACTABLE 0 * /
AT+APPSKEY=?<CR>
<CR>
AT_ERROR<CR>

3.5.5 AT + NWKSKEY - ключ сеанса сети

Описание	Устанавливает / получает ключ сетевой сессии. Этот ключ используется в режимах OTAA и ABP. В режиме OTAA этот ключ заменяется во время процесса создания корневым ключом сети и информацией ответа JoinAccept.
Синтаксис	AT+NWKSKEY=<key><CR> AT+NWKSEY=?<CR>
Аргументы	<key>, 16-байтовое значение, разделенное ":" (строка в шестнадцатеричном формате)
Ответ	<key><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: установить сеансовый ключ NWK * /
AT+NWKSKEY=2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>
/ * Пример 2: получить сеансовый ключ NWK при #define KEY_EXTRACTABLE 1 * /
AT+NWKSKEY=?<CR>
2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C<CR>
<CR>
OK<CR>
/ * Пример 3: получить сеансовый ключ NWK при #define KEY_EXTRACTABLE 0 * /
AT+NWKSKEY=?<CR>
<CR>
AT_ERROR<CR>

3.5.6 AT + DADDR - адрес устройства

Описание	Устанавливает / получает адрес устройства.
Синтаксис	AT+DADDR=<address><CR> AT+DADDR=?<CR>
Аргументы	<address>, 4-байтовое значение, разделенное ":" (строка в шестнадцатеричном формате)
Ответ	<address><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: установить адрес устройства * /
AT+DADDR=01:02:0A:0B<CR>
<CR>
OK<CR>
/ * Пример2: получить адрес устройства * /
AT+DADDR=?<CR>
01:02:0A:0B<CR>
<CR>
OK<CR>

3.5.7 AT + DEUI - Устройство EUI

Описание	Устанавливает / получает устройство EUI.
Синтаксис	AT+DEUI=<EUI><CR> AT+DEUI=?<CR>
Аргументы	<EUI>, 8-байтовое значение, разделенное ":" (строка в шестнадцатеричном формате)
Ответ	<EUI><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: установить EUI устройства * /
AT+DEUI=01:02:03:04:05:06:07:08<CR>
<CR>
OK<CR>
/ * Пример2: получить EUI устройства * /
AT+DEUI=?<CR>
01:02:03:04:05:06:07:08<CR>
<CR>
OK<CR>

3.5.8 AT + NWKID - сетевой идентификатор

Описание	Устанавливает / получает идентификатор сети.
Синтаксис	AT+NWKID=<id><CR> AT+NWKID=?<CR>
Аргументы	<id>, 1-байтовое десятичное значение от 0 до 127
Ответ	<id><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: установить идентификатор сети * /
AT+NWKID=127<CR>
<CR>
OK<CR>
/ * Пример2: получить идентификатор сети * /
AT+NWKID=?<CR>
127<CR>
<CR>
OK<CR>

3.6 Присоединение и отправка данных в сети LoRa

3.6.1 AT + JOIN - присоединиться к сети LoRa

Описание	Присоединиться к сети LoRa
Синтаксис	AT+JOIN=<mode><CR>
Аргументы	<режим> 0: присоединиться к сети через ABP 1: присоединиться к сети через OTAA
Ответ	+EVT:JOINED или +EVT:JOIN_FAILED
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: Присоединиться к сети по ABP * /
#AT+JOIN=0<CR>
+EVT:JOINED<CR> /* событие: настройка ABP завершена. Готов к запуску Tx */
<CR>
OK<CR>
/ * Пример 2: Присоединение к сети по OTAA (Успешный результат) * /
#AT+JOIN=1<CR>
<CR>
OK<CR>
+EVT:JOINED<CR> /* Событие: OTAA присоединение завершено успешно */

/* Пример 3: Присоединение к сети с помощью OTAA (неудачный результат) */
#AT+JOIN=1<CR>
<CR>
OK<CR>
+EVT:JOIN FAILED<CR> /* Событие: событие сбоя присоединения к OTAA. */
/* Сеть LoRaWAN отключена или ключи не согласованы с конфигурацией сети */

3.6.2 AT + LINKC - запрос проверки ссылки

Описание	Командный запрос MAC-команды проверки совмещенного канала для следующего восходящего канала. Выходная информация DemodMargin и NbGateways предоставляется в расширенных событиях Rx + EVT: RX.
Синтаксис	AT+LINKC<CR>
Аргументы	Нет
Ответ	Нет
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Пример:

/ * Пример: запрос проверки обратной связи для следующего восходящего канала * /
#AT+LINKC<CR>
<CR>
OK<CR>

3.6.3 AT + SEND - отправить данные в сеть LoRa

Описание	Отправляет пакеты приложений с указанным AppPort и полезной нагрузкой в сеть LoRaWAN.
Синтаксис	AT+SEND=<port>:<ack>:<payload><CR>
Аргументы	• <port>: порт приложения для передачи • <ack> - 0: неподтвержденное сообщение - 1: подтвержденное сообщение • <payload>: полезная нагрузка в строках шестнадцате-ричного формата (максимальная длина 242 байта).
Ответ	+EVT:SEND_CONFIRMED
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_DUTYCYLE_RESTRICTED<CR><LF> <CR><LF>AT_NO_NET_JOINED<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF> <CR><LF>AT_CRYPTO_ERROR<CR><LF> <CR><LF>AT_ERROR<CR><LF>

Примеры:

/ * Пример 1: отправить пакет на шлюз в неподтвержденном режиме * /
#AT+SEND=2:0:ABCD<CR>
/* отправить пакет : "ABCD", порт приложения - 2, неподтвержденное сообщение */

<CR>
OK<CR>
/ * Пример2: отправить пакет на шлюз в подтвержденном режиме * /
AT+SEND=10:1:7FFF<CR>
/* отправить пакет: "7FFF", с портом приложения 10, подтвержденное сообщение */
<CR>
OK<CR>
+EVT:SEND_CONFIRMED

3.7 Управление сетью LoRa

3.7.1 AT + VER - Версия прошивки

Описание	Получает версию микропрограммы AT_Slave.
Синтаксис	APP_VERSION: Vx.y.z<CR><LF> MW_LORAWAN_VERSION: Va.b.c<CR><LF> MW_RADION_VERSION: Vd.e.f<CR><LF>
Аргументы	Нет
Ответ	<version><CR><LF>
Код результата	<CR><LF>OK<CR><LF>

Пример:

/ * Пример: получение версий приложения и промежуточного программного обеспечения * /
#AT+VER=?
APP_VERSION: V1.1.0<CR>
MW_LORAWAN_VERSION: V2.3.0<CR>
MW_RADION_VERSION: V1.1.0<CR>
<CR>
OK<CR>

3.7.2 AT + ADR - функция адаптивной скорости передачи данных

Описание	Устанавливает / получает функцию адаптивной скорости передачи данных.
Синтаксис	AT+ADR=<enabled><CR> AT+ADR=?<CR>
Аргументы	<включен> • 0: ADR отключен • 1: ADR включен (по умолчанию)
Ответ	<enabled><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: Отключить ADR * /
#AT+ADR=0<CR> /* Отключить ADR */
<CR>
OK<CR> * модуль возвращает код ошибки команды */

/ * Пример 2: Проверить статус ADR * /
AT+ADR=?<CR>
0<CR> /* модуль возвращает статус ADR */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.3 AT + DR - Скорость передачи данных

Описание	Устанавливает / получает скорость передачи данных.
Синтаксис	AT+DR=<data rate><CR> AT+DR=?<CR>
Аргументы	<скорость передачи данных> в диапазоне [0,1,2,3,4,5,6,7]
Ответ	<data rate><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примечание. Чтобы установить скорость передачи данных, необходимо отключить ADR.

Примеры:

/ * Пример1: Установить скорость передачи данных */
#AT+DR=2<CR>/* Установить скорость передачи данных */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/* Пример 2: */
/* Получить скорость передачи данных с отключенной адаптивной скоростью передачи данных */
#AT+ADR=?<CR>
0<CR>
<CR>
OK<CR>
AT+DR=?<CR>
2<CR> /* модуль возвращает скорость передачи данных */
<CR>
OK<CR>
/ * Пример 3: Получить скорость передачи данных с включенной Adaptive DataRate * /
#AT+ADR=?<CR>
1<CR>
<CR>
OK<CR>
AT+DR=?<CR>
<CR>
AT_ERROR<CR>

3.7.4 AT + BAND - Активный регион

Описание	Устанавливает / получает активный регион.
Синтаксис	AT+BAND=<band><CR> AT+BAND=?<CR>
Аргументы	<диапазон>: номер, соответствующий активному региону 0: AS923 1: AU915 2: CN470 3: CN779 4: EU433 5: EU868 6: KR920 7: IN865 8: US915 9: RU864
Ответ	<band><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: Установить активный регион * /
#AT+BAND=0<CR> /* Установить AS923 в качестве активного региона*/
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример2: Получить активный регион * /
AT+BAND=?<CR>
5:EU868<CR> /* модуль возвращает активный регион */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.5 AT + CLASS - Класс LoRa

Описание	Устанавливает / получает класс LoRa.
Синтаксис	AT+CLASS=<class><CR> AT+CLASS=?<CR>
Аргументы	<класс>: должен быть A, B или C.
Ответ	<class><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_ERROR<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_NO_CLASS_B_ENABLE<CR><LF> <CR><LF>AT_NO_NET_JOINED<CR><LF>

Примеры:

/ * Пример1: установить класс LoRa * /
#AT+CLASS=C<CR> /* Установить класс C на устройстве */
<CR>

OK<CR> /* модуль возвращает код ошибки команды */
/* Пример 2: получить класс LoRa */
AT+CLASS=?<CR>
C<CR> /* модуль возвращает Active Class */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.6 AT + DCS - Настройки рабочего цикла

Описание	Устанавливает / получает настройки рабочего цикла.
Синтаксис	AT+DCS=<dutyCycleEnable><CR> AT+DCS=?<CR>
Аргументы	<включение рабочего цикла> 0: рабочий цикл отключен 1: рабочий цикл включен
Ответ	<dutyCycleEnable><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: Включить рабочий цикл * /
#AT+DCS=1<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример 2: Получить рабочий цикл * /
AT+DCS=?<CR>
1<CR> /* модуль возвращает рабочий цикл */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.7 AT + JN1DL - Задержка соединения в окне Rx 1

Описание	Устанавливает / получает задержку принятия соединения между концом Tx и окном 1 Rx соединения (в мс).
Синтаксис	AT+JN1DL=<delay><CR> AT+JN1DL=?<CR>
Аргументы	<задержка>: значение в мс
Ответ	<delay><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример 1: Установить задержку соединения в окне приема 1 * /
#AT+JN1DL=5000<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример 2: Получить задержку соединения в окне приема 1 * /
AT+JN1DL=?<CR>

5000<CR> /* модуль возвращает задержку соединения в окне приема 1 в мс */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.8 AT + JN2DL - Задержка соединения окна 2 Rx

Описание	Устанавливает / получает задержку между концом Tx и соединением окна 2 Rx (в мс).
Синтаксис	AT+JN2DL=<delay><CR> AT+JN2DL=?<CR>
Аргументы	<задержка>: значение в мс
Ответ	<delay><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример 1: Установить задержку соединения окна 2 Rx*/
#AT+JN2DL=8000<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример 2: Получить задержку соединения окна 2 Rx*/
AT+JN2DL=?<CR>
8000<CR> /* модуль возвращает задержку соединения в окне приема 2 в мс */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.9 AT + RX1DL - Задержка окна Rx 1

Описание	Устанавливает / получает задержку между концом Tx и окном Rx 1 (в мс).
Синтаксис	AT+RX1DL=<delay><CR> AT+RX1DL=?<CR>
Аргументы	<задержка>: значение в мс
Ответ	<delay><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример 1: Установить задержку в окне RX 1 * /
#AT+RX1DL=1500<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример 2: Получить задержку в окне RX 1 * /
AT+RX1DL=?<CR>
1500<CR> /* модуль возвращает задержку в окне RX 1 в мс*/
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.10 AT + RX2DL - Задержка окна Rx 2

Описание	Устанавливает / получает задержку между концом Tx и окном Rx 2 (в мс).
Синтаксис	AT+RX2DL=<delay><CR> AT+RX2DL=?<CR>
Аргументы	<задержка>: значение в мс
Ответ	<delay><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример 1: Установить задержку в окне RX 2 * /
#AT+RX2DL=2500<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример 2: Получить задержку в окне RX 2 * /
AT+RX2DL=?<CR>
2500<CR> /* модуль возвращает задержку в окне RX 2 в мс */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.11 AT + RX2DR - Скорость передачи данных окна Rx 2

Описание	Устанавливает / получает скорость передачи данных окна Rx 2 (0-7 соответствует DR_X).
Синтаксис	AT+RX2DR=<datarate><CR> AT+RX2DR=?<CR>
Аргументы	<данные>: значение в диапазоне [0:15]
Ответ	<datarate><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример 1: Установить скорость передачи данных окна RX 2 * /
#AT+RX2DR=5<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример 2: Получить скорость передачи данных окна RX 2 * /
AT+RX2DR=?<CR>
5<CR> /* модуль возвращает скорость передачи данных окна RX 2 */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.12 AT + RX2FQ - Частота окна Rx 2

Описание	Устанавливает / получает частоту окна Rx 2.
Синтаксис	AT+RX2FQ=<freq><CR> AT+RX2FQ=?<CR>
Аргументы	<частота>: значение в Гц
Ответ	<freq><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример 1: Установить частоту окна RX 2 * /
#AT+RX2FQ=869535000<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример 2: Получить частоту окна RX 2 * /
AT+RX2FQ=?<CR>
869535000<CR> /* модуль возвращает частоту окна RX 2 */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.13 AT + TXP - мощность передачи

Описание	Устанавливает / получает мощность передачи.
Синтаксис	AT+TXP=<TxPow><CR> AT+TXP=?<CR>
Аргументы	<TxPow>: должна находиться в диапазоне активированного региона в диапазоне [0:15].
Ответ	<TxPow><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример 1: Установить мощность передачи * /
#AT+TXP=3<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример 2: Получить мощность передачи * /
AT+TXP=?<CR>
3<CR> /* модуль возвращает мощность передачи */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.7.14 AT + PGSLOT - Слот Ping

Описание	Устанавливает / получает периодичность слота одноадресного пинга.
Синтаксис	AT+PGSLOT=<periodicity><CR> AT+PGSLOT=?<CR>
Аргументы	<периодичность>: периодичность для переда- чи, должна быть в диапазоне [0: 7] Периодичность слота Ping составляет 2 <periodicity> в секундах.
Ответ	<periodicity><CR><LF>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример1: Установить слот для проверки связи * /
#AT+PGSLOT=4<CR> /* Установите периодичность Ping Slot на 2 ^ 4 = 16 секунд. */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/* Пример 2: установка слота для проверки связи */
#AT+PGSLOT=?<CR>
4<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.8 Команды тестирования радио

3.8.1 AT + TTONE - тест радиочастотного тона

Описание	Запускает тест радиочастотного тона.
Синтаксис	AT+TTONE<CR>
Аргументы	Нет
Ответ	Нет
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Пример:

/ * Пример: запускает тест RF Tone * /
AT+TTONE<CR>
[TimeDisplay]: Tx FSK Test<CR>
<CR>
OK<CR>

3.8.2 AT + TRSSI - тест тонального сигнала RF RSSI

Описание	Запускает тональный тест RF RSSI.
Синтаксис	AT+TRSSI<CR>
Аргументы	Нет
Ответ	Нет
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Пример:

/ * Пример: запускает тональный тест RSSI * /
AT+TRSSI<CR>
[TimeDisplay]: Rx FSK Test<CR>
[TimeDisplay]:>>> RSSI Value= -7 dBm<CR>
<CR>
OK<CR>

3.8.3 AT + TCONF - конфигурация теста LoRa RF

Описание	Устанавливает / получает конфигурацию теста LoRa RF.
Синтаксис	AT+TCONF=<freq>:<pow>:<bw>:<sf>:<cr>:<lna>:<pa>:<mod>:<paylen>: <freqdev>:<lowdropt>:<BT><CR> AT+TCONF=?<CR>
Аргументы	• <freq>: частота в Гц • <pow>: мощность передачи в диапазоне [-9: 22] дБм • <bw>: - LoRa (в кГц) · 0: 7.8125 · 1: 15.625 · 2: 31.25 · 3: 62.5 · 4: 125 · 5: 250 · 6: 500 - FSK: от 4800 до 467000 Гц • <sf>: - LoRa: от SF5 до SF12 бит / с - FSK: от 600 до 300000 бит / с • <cr>: только LoRa – 1: 4/5 – 2: 4/6 – 3: 4/7 – 4: 4/8 • <lna>: малошумящий усилитель – 0: Off – 1: On

Аргументы	• <pa>: усиление PA – 0: Off – 1: On • <mod>: модуляция – [0: FSK – 1: LoRa – 2: BPSK(Tx) • <paylen>: длина полезной нагрузки от 1 до 256 • <freqdev>: только FSK от 4800 до 467000 • <lowdropt>: оптимизация с низким DR, только LoRa – 0: Off – 1: On – 2: Авто (1 для SF11 или SF12, 0 в противном случае) • <BT>: только FSK - 0: фильтр Гаусса не применяется – 1: BT = 0,3 – 2: BT = 0,5 – 3: BT = 0,7 – 4: BT = 1
Ответ	• Частота = <freq> Hz <CR> • Мощность = <pow> дБм <CR> • Полоса пропускания = <bw> (= 125000 Гц) <CR> • SF = <sf> <CR> • CR = <cr> (= 4/5) <CR> • Состояние LNA = <lna> <CR> • Состояние усиления PA = <pa> <CR> • Модуляция <mod> <CR> • Полезная нагрузка len = <paylen> Байт <CR> • <freqdev> <CR> • LowDRopt [от 0 до 2] = <lowdropt> <CR> • <BT> <CR>
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/ * Пример 1: Установить конфигурацию теста LoRa RF * /
#AT+TCONF=868000000:14:4:12:4/5:0:0:1:16:25000:2:3<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */
/ * Пример 2: Получить конфигурацию теста LoRa RF * /
AT+TCONF=?<CR>
1: Freq= 868000000 Hz<CR>
2: Power= 14 dBm<CR>
3: Bandwidth= 4 (=125000 Hz)<CR>
4: SF= 12<CR>
5: CR= 1 (=4/5)<CR>

6: LNA State= 0<CR>
7: PA Boost State= 0<CR>
8: modulation LORA<CR>
9: Payload len= 16 Bytes<CR>
10: Frequency deviation not applicable<CR>
11: LowDRopt[0 to 2]= 2<CR>
12 BT product not applicable<CR>
can be copy/paste in set cmd: AT+TCONF=868000000:14:4:12:4/5:0:0:1:16:25000:2:3<CR>
<CR>
OK<CR>

3.8.4 AT + TTX - Пакеты, отправляемые для проверки PER RF TX

Описание	Запускает тест PER RF TX с количеством пакетов для отправки.
Синтаксис	AT+TTX=<nb_packets><CR>
Аргументы	<nb_packets>
Ответ	Нет
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Пример:

/ * Пример: запускает тест PER RF TX с количеством пакетов, которые нужно отправить. * /
AT+TTX=4<CR>
[TimeDisplay]:Tx Test<CR>
[TimeDisplay]:Tx Test: Packet 1 of 4<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Test: Packet 2 of 4<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Test: Packet 3 of 4<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Test: Packet 4 of 4<CR>
[TimeDisplay]:OnTxDone<CR>
<CR>
OK<CR>

3.8.5 AT + TRX - пакеты, которые должны быть получены для теста PER RF RX

Описание	Запускает тест PER RF RX с количеством пакетов, которые должны быть получены.
Синтаксис	AT+TRX=<nb_packets><CR>
Аргументы	<nb_packets>
Ответ	Нет
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Пример:

/ * Пример: запускает тест PER RF RX с количеством пакетов, которые должны быть получены. * /
AT+TRLRA=4<CR>
[TimeDisplay]:PRE OK<CR>
[TimeDisplay]:HDR OK<CR>
[TimeDisplay]:OnRxDone<CR>
[TimeDisplay]:RssiValue=-7 dBm, SnrValue=7<CR>
[TimeDisplay]:Rx: 1 of 4 >>> PER= 0 %<CR>
/* Процент PER обновляется / отображается после каждого приема */
[TimeDisplay]:PRE OK<CR>
[TimeDisplay]:HDR OK<CR>
[TimeDisplay]:OnRxDone<CR>
[TimeDisplay]:RssiValue=-7 dBm, SnrValue=6<CR>
[TimeDisplay]:Rx: 2 of 4 >>> PER= 0 %<CR>
/* Процент PER обновляется / отображается после каждого приема */
[TimeDisplay]:PRE OK<CR>
[TimeDisplay]:HDR OK<CR>
[TimeDisplay]:OnRxDone<CR>
[TimeDisplay]:RssiValue=-7 dBm, SnrValue=5<CR>
[TimeDisplay]:Rx: 3 of 4 >>> PER= 0 %<CR>
/* Процент PER обновляется / отображается после каждого приема */
[TimeDisplay]:PRE OK<CR>
[TimeDisplay]:HDR OK<CR>
[TimeDisplay]:OnRxDone<CR>
[TimeDisplay]:RssiValue=-7 dBm, SnrValue=6<CR>
[TimeDisplay]:Rx: 4 of 4 >>> PER= 0 %<CR>
/* Процент PER обновляется / отображается после каждого приема */
<CR>
OK<CR>

3.8.6 AT + TTH - тест скачкообразной перестройки RF Tx

Описание	Запускает тест скачкообразной перестройки RF Tx от Fstart до Fstop с шагами Fdelta.
Синтаксис	AT+TTH=<Fstart>,<Fstop>,<FDelta>,<nb_packets><CR>
Аргументы	• <Fstart>: начало частоты (в Гц или МГц) • <Fstop>: конц частоты (в Гц или МГц) • <Delta>: полоса частот (в Гц) • <nb_packets>: количество пакетов для отправки
Ответ	Нет
Код результата	<CR><LF> OK <CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF> <CR><LF>AT_BUSY_ERROR<CR><LF>

Пример:

/ * Пример: установить тест скачкообразной передачи TX от 868 до 868,5 МГц с 6 шагами по 100 кГц * /
AT+TTH=868000000,868500000,100000,6<CR>
[TimeDisplay]: Tx Hop at 868000000Hz. 0 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868100000Hz. 1 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868200000Hz. 2 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868300000Hz. 3 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868400000Hz. 4 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
[TimeDisplay]:Tx Hop at 868500000Hz. 5 of 6<CR>
[TimeDisplay]:Tx LoRa Test<CR>
[TimeDisplay]:Tx 1 of 1<CR>
[TimeDisplay]:OnTxDone<CR>
<CR>
OK<CR>

3.8.7 AT + CERTIF - Модуль в сертификации LoRaWAN с режимом соединения

Описание	Запускает модуль в сертификации LoRaWAN и с выбором режима присоединения.
Синтаксис	AT+CERTIF=<mode><CR>
Аргументы	<режим> 0: присоединиться к сети через ABP 1. присоединиться к сети через OTAA
Ответ	+EVT:JOINED +EVT:JOIN_FAILED
Код результата	<CR><LF>OK<CR><LF> <CR><LF>AT_PARAM_ERROR<CR><LF>

Примеры:

/* Пример 1: установить модуль в сертификации LoRaWAN */
/* и присоединиться к сети с помощью ABP */
#AT+CERTIF=0<CR>

+EVT:JOINED<CR> /* событие: настройка ABP завершена. Готов к запуску Tx */
<CR>
OK<CR>
/ * Пример 2: установить модуль в сертификации LoRaWAN и присоединиться к сети по OTAA * /
#AT+CERTIF=1<CR>
<CR>
OK<CR>
+EVT:JOINED<CR> /* Событие: OTAA присоединение успешно */

3.8.8 AT + TOFF - RF тест

Описание	Останавливает RF тест.
Синтаксис	AT+TOFF<CR>
Аргументы	Нет
Ответ	Нет
Код результата	<CR><LF>OK<CR><LF>

Пример:

/ * Пример: останавливает RF тест * /
AT+TOFF<CR>
Test Stop<CR>
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

3.9 Информация

3.9.1 AT + BAT - Уровень заряда батареи

Описание	Получает уровень заряда батареи (в мВ).
Синтаксис	AT+BAT=?<CR>
Аргументы	Нет
Ответ	<уровень> <CR> <LF>: значение в мВ
Код результата	<CR><LF>OK<CR><LF>

Пример:

/ * Пример: получить уровень заряда батареи в мВ * /
#AT+ BAT=?<CR>
3300<CR> /* уровень заряда батареи в мВ */
<CR>
OK<CR> /* модуль возвращает код ошибки команды */

4 Примеры

Вот несколько основных примеров, описывающих, как использовать AT-команды. В следующих разделах командам, предоставляемым хостом, предшествует символ #, а комментарии заключаются в / * * /.