CMOSTEK AN239

Usage Guide for CMT2310A Auto-transceiver Function

Overview

This document introduces the configuration and usage guide for CMT2310A auto-transceiver function.

The part numbers covered in this document are shown in the table below.

Table 1. Part Number Covered in This Document

Part Number	Frequency Range	Modulation Method	Chip Function	Configuration Method	Package
CMT2310A	113 - 960MHz	(4)(G)FSK/OOK	Transceiver	Register	QFN24

Before reading this document, please read the *AN237 CMT2310A Quick Start Guide* first to get the basic usage information of the CMT2310A.

CMOSTEK Confidential

Table of Contents

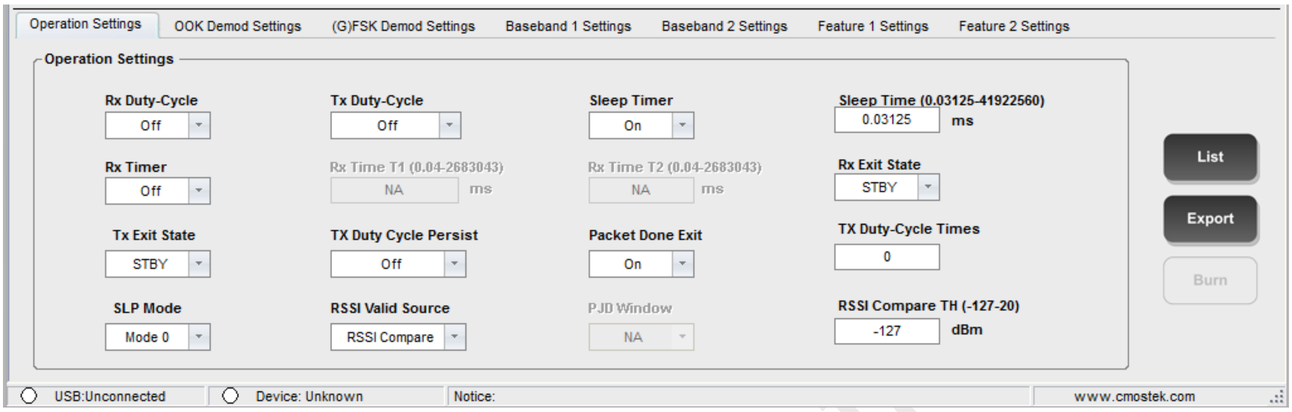
CMOSTEK Confidential

CMOSTEK Confidential

1. Duty-Cycle Operating Mode

1.1 Duty-Cycle Mode Related Register

The corresponding RFPDK GUI and parameters are as follow.



Operation Settings

Rx Duty-Cycle: Off
Tx Duty-Cycle: Off
Sleep Timer: On
Sleep Time (0.03125-41922560): 0.03125 ms
Rx Timer: Off
Rx Time T1 (0.04-2683043): NA ms
Rx Time T2 (0.04-2683043): NA ms
Rx Exit State: STBY
Tx Exit State: STBY
TX Duty Cycle Persist: Off
Packet Done Exit: On
TX Duty-Cycle Times: 0
SLP Mode: Mode 0
RSSI Valid Source: RSSI Compare
PJD Window: NA
RSSI Compare TH (-127-20): -127 dBm

○ USB:Unconnected ○ Device: Unknown Notice: www.cmostek.com

Figure 1. Duty-Cycle GUI on RFPDK

Table 1. Duty-Cycle Related Parameters

Register Bit RFPDK Parameter	Register Bit
Rx Duty-Cycle	RX_DC_EN
Tx Duty-Cycle	TX_DC_EN
Sleep Timer	TIMER_SLEEP_EN
Sleep Time	TIMER_M_SLEEP<10:0> TIMER_R_SLEEP<3:0>
Rx Timer	TIMER_RX_EN
Rx Time T1	TIMER_M_RX_T1<10:0> TIMER_R_RX_T1<3:0>
Rx Time T2	TIMER_M_RX_T2<10:0> TIMER_R_RX_T2<3:0>
Rx Exit State	RX_EXIT_STATE<1:0>
Tx Exit State	TX_EXIT_STATE<1:0>
TX Duty Cycle Persist	TX_DC_PERSIST_EN
Packet Done Exit	PKT_DONE_EXIT_EN
TX Duty-Cycle Times	TX_DC_TIMES<7:0>
SLP Mode	SLP_MODE<3:0>
RSSI Valid Source	CCA_INT_SEL<1:0>
PJD Window	PJDET_WIN<1:0>
RSSI Compare TH	RSSI_ABSOLUTE_TH<7:0>

CMOSTEK Con

1.2 RX in Duty-Cycle Mode

The related register position and description in RX Duty-Cycle are shown as follow:

Table 2. Registers in Configuration Area

Register Name	Bits	R/W	Bit Name	Function Description
CTL_REG_97 (0x61)	6:4	RW	RX_EXIT_STATE<2:0>	Exit to preset state after transmission completes： 1：SLEEP 2：READY 3：TFS 4：TX 5：RFS 6：RX Others：SLEEP It will automatically exit RX in packet mode. Otherwise, the chip will wait until the MCU sends go_* command to switch.
	3	RW	TIMER_RX_EN	RX TIMER enabling： 0：disable 1：enable
	0	RW	RX_DC_EN	RX Duty Cycle enabling： 0：disable 1：enable
CTL_REG_99 (0x63)	7:0	RW	TIMER_M_SLEEP<7:0>	The counting time of SLEEP TIMER is defined as the formula of： T = M x 2^(R+1) x 31.25 us The value range of R is 0-26.
CTL_REG_100 (0x64)	7:5	RW	TIMER_M_SLEEP<10:8>	
	4:0	RW	TIMER_R_SLEEP<4:0>	
CTL_REG_101 (0x65)	7:0	RW	TIMER _M_ RX_T1<7:0>	The counting time of RX T1 TIMER is defined as the formula of： T = M x 2^(R+1) x 20 us The value range of R is 0-21.
CTL_REG_102 (0x66)	7:5	RW	TIMER_M_ RX_T1<10:8>	
	4:0	RW	TIMER _R_RX_T1<4:0>	
CTL_REG_103 (0x67)	7:0	RW	TIMER_M_RX_T2<7:0>	The counting time of RX T2 TIMER is defined as the formula of： T = M x 2^(R+1) x 20 us The value range of R is 0-21.
CTL_REG_104 (0x68)	7:5	RW	TIMER_M_RX_T2<10:8>	
	4:0	RW	TIMER_R_RX_T2<4:0>	
CTL_REG_105 (0x69)	3	RW	TIMER_SLEEP_EN	SLEEP TIMER enabling： 0：disable 1：enable

CMOSTEK Confidential

It needs 4 key register control bits to control RX in Duty-Cycle mode. Coordinated with practical requirements, there are 5 combinations of the 4 control bits and the corresponding operating modes are listed as follow.

In the following state operating diagram, gray lines represents that it requiring sending go_* command by MCU manually to switch state; blue lines represents that the chip can switch state automatically. It should be noted that, when switching from state A to state B, if blue lines already exist, then the gray lines should not appear and vice versa. That is, if it adopts auto-control already, manual control by MCU is not allowed. Otherwise, it will disrupt the operation of the chip and even crash the system. It is the same when manual switching adopted, auto-switching won't be turned on.

CMOSTEK Confidential

1.2.1 Full-manual Control

Full-manual control means no Duty-Cycle control is enabled. All the state switching is accomplished through software by users.

Table 3. Full Manual Control

Control Bit	Value
TIMER_SLEEP_EN	0
RX_DC_EN	0
TIMER_RX_EN	0
RX_EXIT_STATE	0

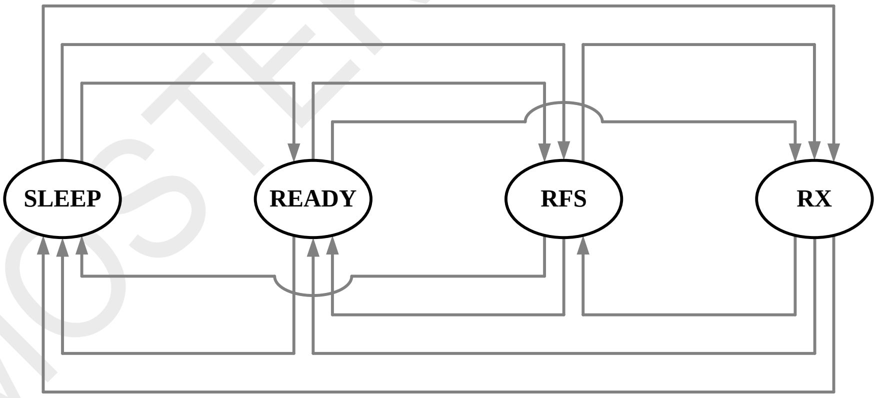


Figure 2. Full-manual Control of the Receiver

1.2.2 Automatic Sleep Wake-up

In this mode, only SLEEP TIMER is enabled and the sleep time is defined by users. After being woken up, the chip will switch to READY state and wait for operation of the MCU. According to above operating principles, once the chip enters SLEEP mode, MCU can no longer send command to skip out from SLEEP and it can only wait until the sleep timer timeout, namely it can start only after interruption of SLEEP_TIMEOUT_FLAG detected.

Table 4. Automatic SLEEP Wake-up

Control Bit	Value
TIMER_SLEEP_EN	1
RX_DC_EN	0
TIMER_RX_EN	0
RX_EXIT_STATE	0

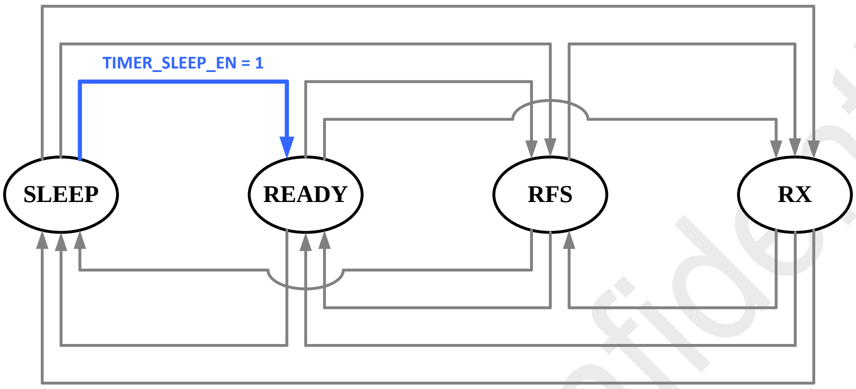


Figure 3. Automatic SLEEP Wake-up of Receiver

1.2.3 Automatic SLEEP Wake-up, Automatic RX Entry

When RX_DC_EN is enabled, it will not skip to READY after the chip is awaken from SLEEP automatically; instead, it will enter PLL calibration directly and switch to RX to receive. In this mode, the operations of the MCU are greatly reduced.

Table 5. Automatic SLEEP Wake-up, Automatic RX Entry

Control Bit	Value
TIMER_SLEEP_EN	1
RX_DC_EN	1
TIMER_RX_EN	0
RX_EXIT_STATE	0

CMOSTEK Confidential

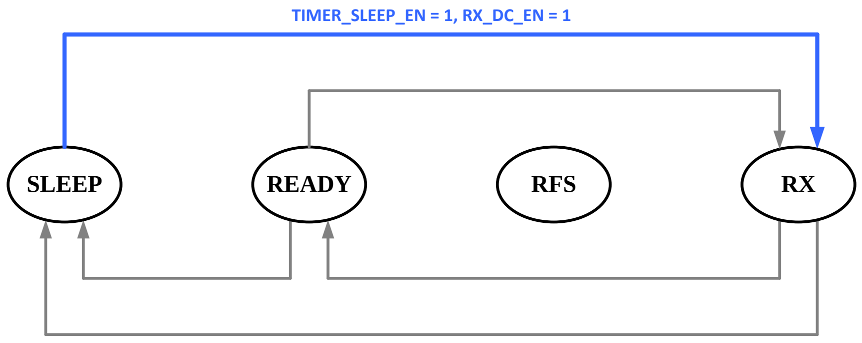


Figure 4. Automatic SLEEP Wake-up, Automatic RX Entry for Receiver

CMOSTEK Confidential

1.2.4 Automatic SLEEP Wake-up, Automatic RX Exit

In this mode, with function of automatic SLEEP wake-up and automatic exiting RX., it still needs MCU to participate in manual switching when entering RFS or RX from READY. (namely, it will not receive automatically). However, once it enters RX, RX TIMER will begin to count and exit upon timeout and automatically switch to the corresponding state configured through RX_EXIT_STATE. In general, READY state can be treated as a 'transferring station', which allows the MCU to participate in operations such as clearing interrupt and reading FIFO.

Table 6. Automatic SLEEP Wake-up, Automatic RX Exit

Control Bit	Value
TIMER_SLEEP_EN	1
RX_DC_EN	0
TIMER_RX_EN	1
RX_EXIT_STATE	1/2/5

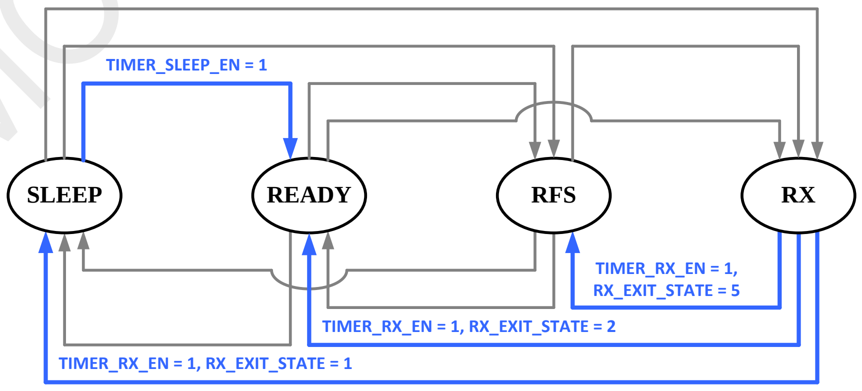

TIMER_SLEEP_EN = 1
SLEEP
READY
RFS
RX
TIMER_RX_EN = 1,
RX_EXIT_STATE = 5
TIMER_RX_EN = 1, RX_EXIT_STATE = 2
TIMER_RX_EN = 1, RX_EXIT_STATE = 1

Figure 5. Automatic SLEEP Wake-up and Automatic RX Exit for Receiver

1.2.5 Fully Automatic RX

Once fully automatic receiving mode enabled, it does not need MCU to participate in state switching (and MCU cannot participate in state switching as well). MCU can only acquire operating state through the preset interrupts and then conduct required operations. It should be noted that, before entering into fully automatic TX, the MCU must set the configurations well in READY state, which includes packet format, FIFO operating mode, interruption as well as IO.

Table 7. Fully Automatic RX

Control Bit	Value
SLEEP_TIMER_EN	1
RX_DC_EN	1
RX_TIMER_EN	1
RX_EXIT_STATE	1

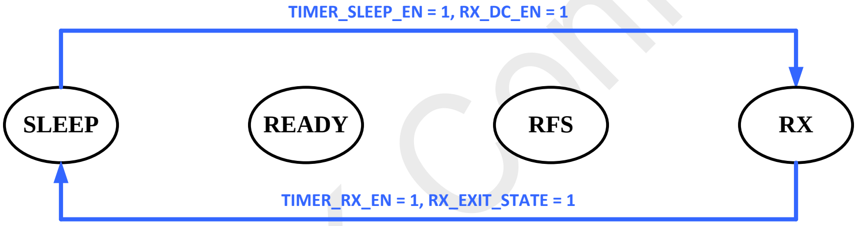

TIMER_SLEEP_EN = 1, RX_DC_EN = 1
SLEEP
READY
RFS
RX
TIMER_RX_EN = 1, RX_EXIT_STATE = 1

Figure 6. Fully Automatic Control of Receiver

This operating mode has a limit that speed of the MCU has to be quick enough and take advantage of interaction between interruption and chip, so as to read FIFO completely during RX.

For collaboration between MCU and CMT2310A, users should pay special attention to follow the operating principle that once automatic control is enabled, it is not allowed to send command manually to have state switching. Furthermore, it should be noted that, once automatic exit RX is enabled, and if the target exit state is SLEEP, then interrupt state information in control area 2 will all be lost after it exits SLEEP state. Therefore, users should have an overall consideration when this mode is applied to make sure MCU can interact with the chip robustly, avoiding the circumstances of failing in waiting and catching interrupts. Generally, we suggest users setting RX_EXIT_STATE as 2, namely staying in READY state after exiting RX automatically. At this point, all the interrupt information can be reserved and the power consumption will reduce a lot; Besides, it can make MCU process and complete all the work and then manually switch the chip to SLEEP state.

CMOSTEK Confidential

1.3 TX Duty-Cycle Mode

The related register description is shown as follow.

Table 8. Tx Duty-Cycle Related Register

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_23 (0x17)	4	RW	TX_DC_DONE_EN	TX_DC_DONE_FLAG and TX_DC_DONE interruption configuration enabling. When corresponding TX_DC_DONE_FLAG event is valid，TX_DC_DONE_FLAG will set as 1 and generate TX_DC_DONE interruption. 1：enable 0：disable
CTL_REG_31 (0x1F)	4	W	TX_DC_DONE_CLR	TX_DC_DONE_FLAG clearing bit 1：Make TX_DC_DONE_FLAG cleared. 0：Have no effect to TX_DC_DONE_FLAG
CTL_REG_32 (0x20)	4	R	TX_DC_DONE_FLAG	TX Duty CLCLE isn't enabled the persistent mode. If it reaches the maximum TX number while TX_DC_DONE_EN enabled，TX_DC_DONE_FLAG will be set as 1 1：tx duty cycle reaches the maximum TX number 0：tx duty cycle hasn't reached the maximum TX number
CTL_REG_96 (0x60)	6:4	RW	TX_EXIT_STATE<2:0>	After TX, the chip will automatically exit to the setting state, which is valid only in packet mode. Otherwise, the chip will not automatically exit TX state, but wait for MCU to send go_* command to switch. 1：SLEEP 2：READY 3：TFS 4：TX 5：RFS 6：RX Others：SLEEP
	1	RW	TX_DC_PERSIST_EN	Configuration of TX duty cycle under TX mode. 0：Exit when the configuration number TX_DC_TIMES completes. 1：Keep going until this bit is configured to 0
	0	RW	TX_DC_EN	TX Duty Cycle enabling 0：disable 1：enable
CTL_REG_99 (0x63)	7:0	RW	TIMER_M_SLEEP<7:0>	Counting time of SLEEP TIMER is defined as the formula of：

CMOSTEK Confidential

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_100 (0x64)	7:5	RW	TIMER_M_SLEEP<10:8>	T = M x 2^(R+1) x 31.25 us The value range of R is 0-26.
	4:0	RW	TIMER_R_SLEEP<4:0>	
CTL_REG_105 (0x69)	3	RW	TIMER_SLEEP_EN	SLEEP TIMER enabling： 0: disable. 1: enable.
CTL_REG_110 (0x6E)	7:0	RW	TX_DC_TIMES<7:0>	The stated maximum TX number of non-persistent under TX Duty Cycle mode.
CTL_REG_112 (0x70)	7:0	R	TX_DC_DONE_TIMES<7:0>	The already complete TX number under TX Duty Cycle mode.

It needs 3 register control bits to control TX Duty-cycle mode. For TX, there are no TX TIMER for the reason that the TX is an active behavior with very high power efficiency and the chip will automatically exit the TX state since the TX complete under the packet mode. Therefore, the TX time is entirely depended on the TX content.

According to practical application requirements, there listed 3 combinations of the 3 control bits and the corresponding operating modes in the following chapter. The control principles are identical to RX Duty-Cycle mode.

CMOSTEK Confidential

1.3.1 Automatic TX Exit

This is the most simply mode that is will switch to the state specified in TX_EXIT_STATE automatically after TX completes. The other cases need to manually control by MCU.

Table 9. Automatically Exit TX

Control Bit	Value
TIMER_SLEEP_EN	0
TX_DC_EN	0
TX_EXIT_STATE	1/2/3

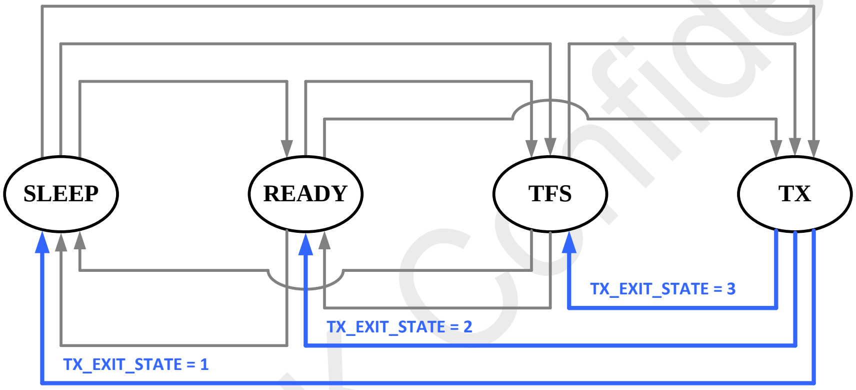


Figure 7. Automatic TX Exit of Transmitter

1.3.2 Automatic SLEEP Wake-up, Automatic TX Exit

In this mode, not only it can exit TX automatically, but also open the SLEEP TIMER. The chip will switch to READY state until MCU has further operation after the automatic wake-up.

Table 10. Automatic SLEEP Wake-up, Automatic TX Exit

Control Bit	Value
TIMER_SLEEP_EN	1
TX_DC_EN	0
TX_EXIT_STATE	1/2/3

CMOSTEK Confidential

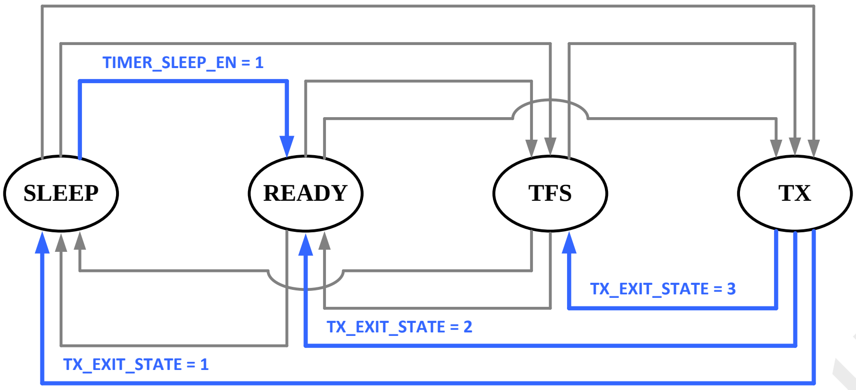


Figure 8. Transmitter wake up from Sleep mode and exit TX automatically

1.3.3 Fully Automatic TX

Once fully automatic TX enabled, it does not need MCU to participate in state switching (and MCU cannot participate in state switching as well). MCU can only acquire operating state through the preset interrupts and then conduct required operations. It should be noted that, before entering into fully automatic TX, the MCU must set the configurations well in READY state, which includes packet format, FIFO operating mode, interruption as well as IO.

CMOSTEK Confidential

Table 11. Fully Automatic TX

Control Bit	Value
TIMER_SLEEP_EN	1
TX_DC_EN	1
TX_EXIT_STATE	1

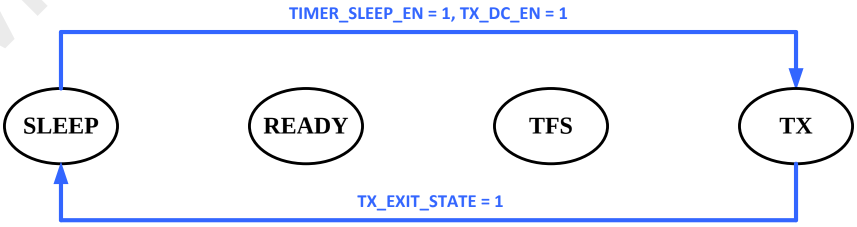


Table 9. Fully Automatic TX of Transmitter

1.4 Duty-Cycle Mode Exit and Entry

1.4.1 Duty-Cycle Mode Entry

After chip initialization completes, it goes to the configuration stage within which users can configure related registers to enter the needed Duty-Cycle mode. The configuration must be conducted in READY state and then enters SLEEP/TX/RX manually. Therefore, the chip starts to operate according to the configuration.

1.4.2 Duty-Cycle Mode Exit

If it is not in fully automatic Duty-Cycle mode, the system will stay in a certain state and wait for operations from MCU. MCU can switch the system back to READY then reconfigure the related register again according to the configuration flow discussed above and then exit the Duty-Cycle.

If it is in fully automatic Duty-Cycle mode, no matter TX or RX, MCU does not know the exact chip operating state. So that, a 100% reliable mechanism is needed to let MCU stop the automatic operating and switch back to manual control mode.

If the CMT2310A starts automatic Duty-Cycle operating after initialization configuration, then MCU needs to operate the registers below to exit regularly.

1. Set API_STOP to 1, then the chip will return to READY state upon next state switching.
2. MCU can keep on scanning register CTL_REG_MODE<7:0> until it confirms the chip enters READY state.
3. Reconfigure Duty-Cycle related registers and close fully automatic mode (can configure other registers as well). After configuration done, set API_STOP to 0.
4. Send go_sleep command to make the configuration effectively. Then the system will stay in SLEEP state until further operation of MCU.

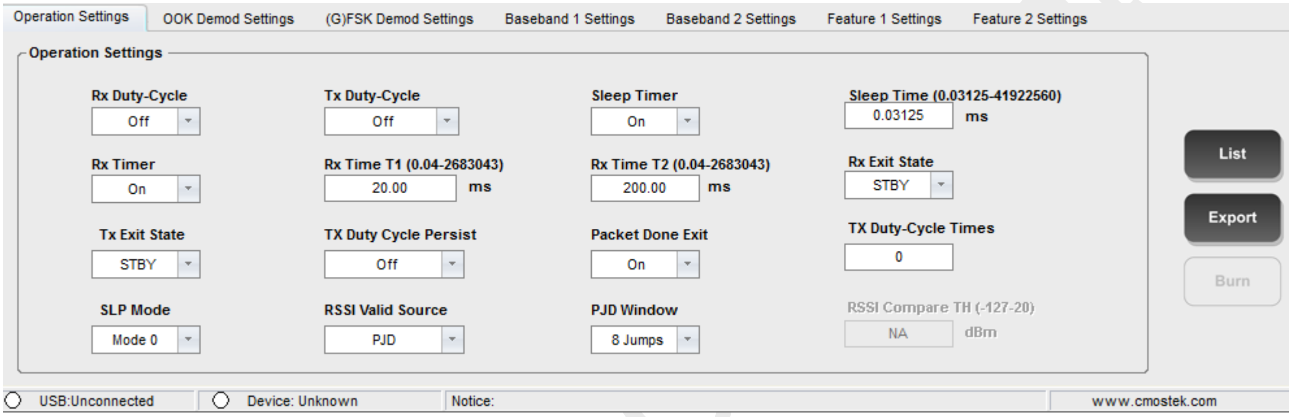
CMOSTEK Confidential

2. Super-low Power RX Mode

CMT2310A provide a series of options which can help users to fulfill super-low power (SLP) RX according to different application requirements. All these options can take effect only when RX_TIMER_EN is set to 1, namely RX timer is enabled. The core of SLP RX is about how to reduce RX time of the receiver best when there is no signal, and how to extend the RX time for receiving properly, then to achieve stable receiving and minimum power consumption. The SLP mode belongs to automatic mode and it will stay in a certain state waiting for the operation of external MCU under certain conditions

2.1 SLP Receiving Related Register

The corresponding RFPDK GUI and parameters are as follow.



Operation Settings

Rx Duty-Cycle: Off
Tx Duty-Cycle: Off
Sleep Timer: On
Sleep Time (0.03125-41922560): 0.03125 ms
Rx Timer: On
Rx Time T1 (0.04-2683043): 20.00 ms
Rx Time T2 (0.04-2683043): 200.00 ms
Rx Exit State: STBY
Tx Exit State: STBY
TX Duty Cycle Persist: Off
Packet Done Exit: On
TX Duty-Cycle Times: 0
SLP Mode: Mode 0
RSSI Valid Source: PJD
PJD Window: 8 Jumps
RSSI Compare TH (-127-20): NA dBm

Figure 10. SLP Related RFPDK GUI

SLP mode is based on RX Duty Cycle mode, so registers related to RX Duty Cycle still need to be configured in practical application. Except for registers related to RX Duty Cycle, parameters related to SLP register are shown as follows

Table 12. SLP Related Register

RFPDK Parameters	Register Bits
Packet Done Exit	PKT_DONE_EXIT_EN
SLP Mode	SLP_MODE<3:0>

The related registers are described in the table below.

Table 13. SLP Related Register

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_98 (0x62)	7	RW	PKT_DONE_EXIT_EN	Whether to keep the current state or exit immediately and return to the corresponding RX_EXIT_STATE state when

CMOSTEK Confidential

Register Name	Bit	R/W	Bit Flag	Function Description
				the chip receives packet successfully and generates PKT_DONE. 0：The chip remains in the current state 1：The chip returns to the corresponding state according to the RX_EXIT_STATE configuration.
	3:0	RW	SLP_MODE<3:0>	RX Duty Cycle Configuration The valid range is 0-13. Specific information please see at Table 17. the SLP Mode Description.

2.2 Principle of Low-power RX and TX

In traditional short-distance wireless RX and TX system, the basic low-power RX and TX solutions are as follows. CMT2310A not only is compatible to all these solutions but expands 14 more quality power-saving solutions. The basic solutions will be discussed first below, which can be fulfilled through setting SLP_MODE <3:0> to 0.

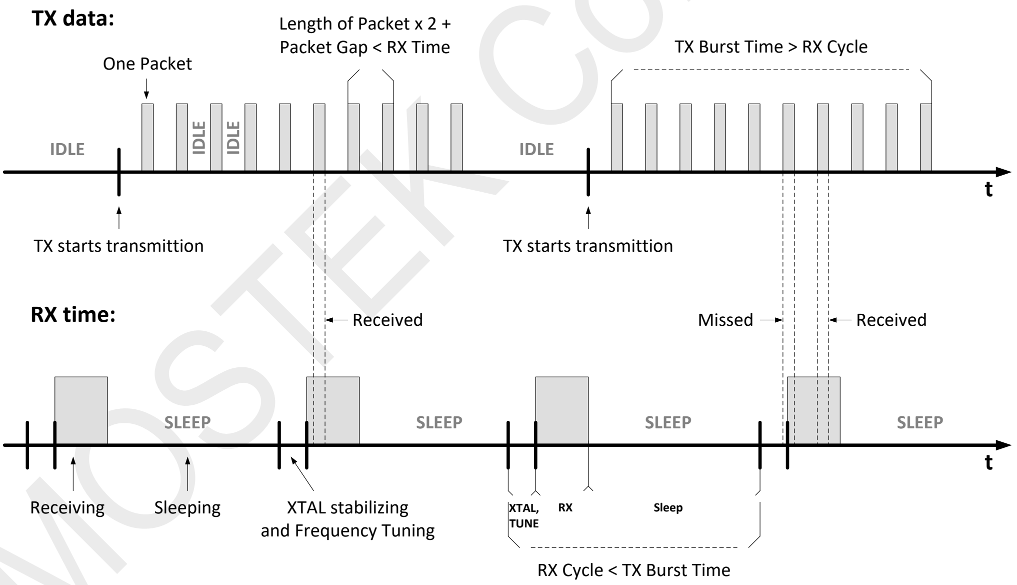


Figure 11. Basic Low-power RX and TX Solution

CMOSTEK Confidential

As shown in the above figure, as long as the two calculation relations below are satisfied, it can definitely catch the TX data in RX Duty-Cycle Mode.

1. The entire RX cycle < the total time of sending N packets in a batch at TX side
2. RX time > 2 packets + 1 packet interval

Among which, an entire RX cycle = RX time + Sleep time + Crystal Oscillation startup and settling time + PLL frequency calibration time.

It can be seen that by using this basic low-power scheme, constrained by computational relations, users first need to make a compromise between the time of SLEEP and the length of transmitted data, that is, whether to save more power at RX side or at TX side Second, users must set the RX time window large enough to capture 100% of the data.

2.3 Signal Channel Sensing

Before discussing SLP mode, we will introduce the important assisting mechanism of SLP, signal channel sensing. In this mechanism, it senses whether the valid signal is appearing to generate the signal RSSI_PJD_VLD (1 represents signal appeared and 0 represents noise). This signal will not only be output to GPIO but act as a triggering condition to assist the SLP implementation.

There are two kinds of signal channel sensing mechanism, Phase Jump Detector (PJD) and RSSI comparison.

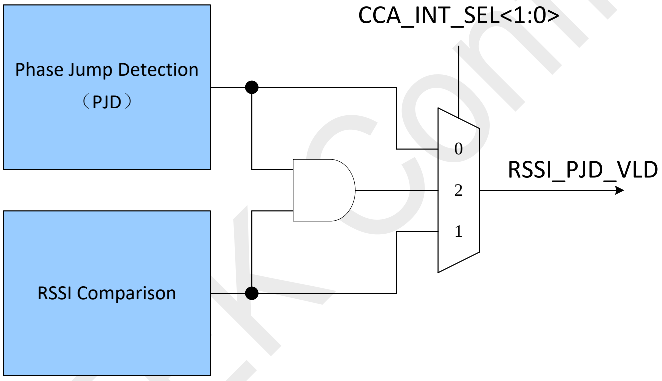


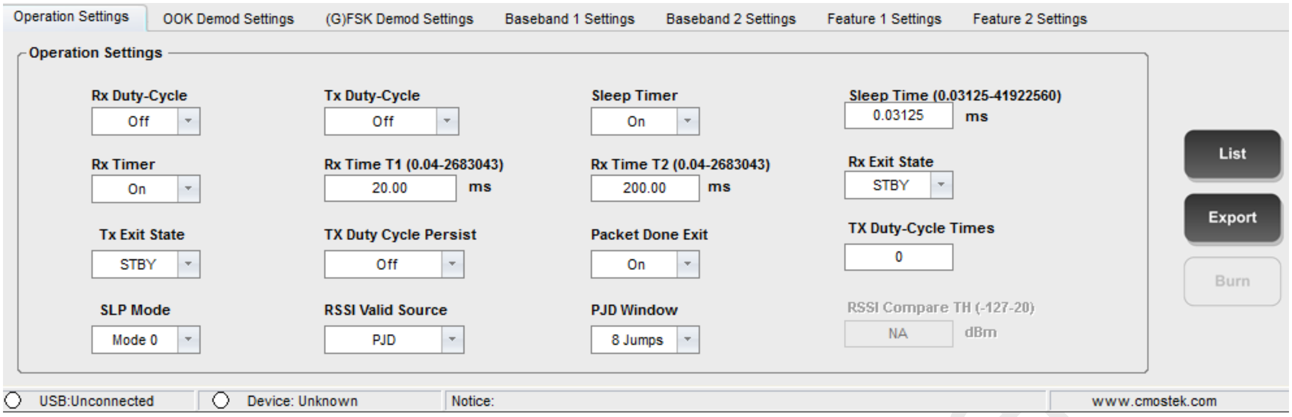
Figure 12. Signal Channel Sensing Mechanism

It should be noted that, PJD can only be used in 2-FSK mode, while RSSI comparison can be used all in 2-FSK, 4-FSK and OOK modes.

CMOSTEK Confidential

2.3.1 Signal Channel Sensing Related Register

The corresponding RFPDK GUI and parameters are as follows.



Operation Settings | OOK Demod Settings | (G)FSK Demod Settings | Baseband 1 Settings | Baseband 2 Settings | Feature 1 Settings | Feature 2 Settings

Operation Settings

Rx Duty-Cycle: Off
Tx Duty-Cycle: Off
Sleep Timer: On
Sleep Time (0.03125-41922560): 0.03125 ms

Rx Timer: On
Rx Time T1 (0.04-2683043): 20.00 ms
Rx Time T2 (0.04-2683043): 200.00 ms
Rx Exit State: STBY

Tx Exit State: STBY
TX Duty Cycle Persist: Off
Packet Done Exit: On
TX Duty-Cycle Times: 0

SLP Mode: Mode 0
RSSI Valid Source: PJD
PJD Window: 8 Jumps
RSSI Compare TH (-127-20): NA dBm

Figure 13. Signal Channel Sensing RFPDK GUI

Table 14. Signal Channel Sensing Related Parameters

Register-bit RFPDK Parameter	Register bit
RSSI Valid Source	CCA_INT_SEL<1:0>
PJD Window	The parameter defines how many jumps is required to judge whether it is signal or noise. 0: 4 times 1: 6 times 2: 8 times 3:10 times
RSSI Compare TH	RSSI valid signal generated when the value is higher than the RSSI comparison threshold.

The register content and description is listed in the table below.

Table 15. Registers in Configuration Area

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_106 (0x6A)	3:2	RW	CCA_INT_SEL<1:0>	Condition of RSSI_PJD_VALID interrupt generated： 00：PJD is valid 01：RSSI is valid 10：Both PJD and RSSI are valid 11：NA

CMOSTEK Confide

2.3.2 RSSI Comparison

This feature will be described in detail at the following section on RSSI measurement and Comparison, however here only shows a brief introduction. The principle of RSSI comparison is that, if the RSSI of signal or noise is higher than the threshold, then RSSI_VLD will be valid; otherwise it is invalid. This mechanism advances in the applying of both FSK and OOK and the disadvantage is that the threshold setting needs to be adjusted according to the actual application environment, and should be avoided to be triggered by noise and interference signals. This mechanism has no apparent advantage in assisting SLP implementation through RSSI_VLD generation. Therefore, we focus more on the PJD mechanism discussed below.

2.3.3 Phase Jump Detection（PJD）

PJD is a new technology observing the jump characteristics of the received signal in the chip FSK demodulation to determine whether the received signal is a noise or useful signal. PJD believes that the input signal from 0 to 1 or from 1 to 0 switch is a phase jump, users only need to configure PJD_WIN_SEL<1:0> to tell PJD how many times the signal jump should be detected before the judged result being output.

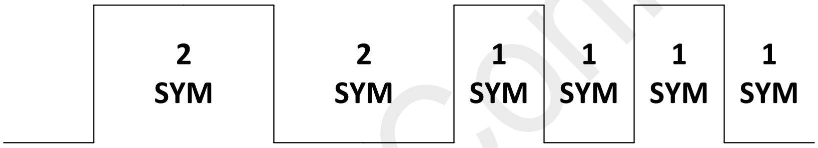


Figure 14. Received Signal Jump Diagram

As shown in figure above, 8 symbols are received with 6 jumps, so the jump number is not equal to total symbol number. The jump number is equal to symbol number only when preambles are received, which should be carefully paid attention.

It will be discussed in the section below how signal RSSI_VLD helps to implement SLP RX mode. In summary, the more the PJD jump times the more reliable the judge result, The less it is, the quicker to complete the judge. If the receiving time window is quite small, then it requires less detection times to meet the time window setting requirement.

Based on practical test data, in general, 4 jumps are enough to get reliable detection result, that is, neither a noise signal judged as useful signal nor useful signal fails to be detected.

2.4 SLP Receiving Mode Description

As described above, the core of SLP is to control the RX time to achieve the purpose: when there is no useful signal at ordinary times, the RX time is very short and it is only used to detect the useful signals; When a useful signal arrives, RX takes longer to successfully receive the required packet.

Therefore, SLP has further control of the RX state on the basis of various manual, semi-automatic and fully automatic RX duel-cycle control modes mentioned above, namely the SLP_MODE<3:0>, which defines 14 SLP modes to control the RX time. Besides the control of RX state, whether automatic or manual, is not necessarily related to SLP itself. Users can implement different SLP modes in different RX duty-cycle modes

CMOSTEK Confidential

The 14 SLP modes will be explained in detail below. First, assuming that the CMT2310A works in RX duty-cycle mode, automatically wakes up from SLEEP, automatically exits from RX and switches to SLEEP. All 14 SLP modes have evolved on this basis.

Table 16. RX Duty-Cycle Mode in SLP Example

Control Bit	Value
SLEEP_TIMER_EN	1
RX_DC_EN	1
RX_TIMER_EN	1
RX_EXIT_STATE	1

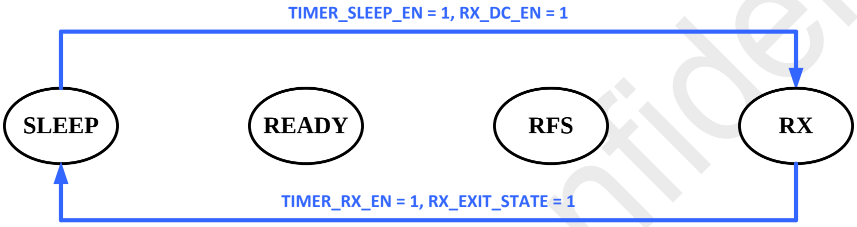

TIMER_SLEEP_EN = 1, RX_DC_EN = 1
SLEEP
READY
RFS
RX
TIMER_RX_EN = 1, RX_EXIT_STATE = 1

Figure 15. RX Duty-Cycle Mode in SLP Example

Various SLP modes will be introduced basing on the RX Duty-Cycle mode. Here is a summary of the 14 SLP modes. T1 and T2 mentioned in the table refer to time windows of the RX T1 and T2 which can be set with registers respectively.

CMOSTEK Confidential

Table 17. Description of SLP Modes

No.	RX Extension Method	RX Extension Condition
0	If it is configured as 0 with no extension, it will exit RX when T1 timing ends.	None
1	Once the detection conditions are met in T1, it leaves T1 and hands over the control to MCU	RSSI_PJD_VALID being valid.
2		PREAM_OK being valid.
3		Both RSSI_VLD and PREAM_OK being valid at the same time.
4	As long as it detects that RSSI is valid in T1, it exits T1 and remains in RX until RSSI is not satisfied and exits RX	RSSI_PJD_VALID being valid.
5	Once the detection conditions are met in T1, it will switch to T2 and exit RX after T2 timing ends	RSSI_PJD_VALID being valid.
6		PREAM_OK being valid.
7		Both RSSI_PJD_VALID and PREAM_OK being valid at the same time.

No.	RX Extension Method	RX Extension Condition
8		Either PREAM_OK or SYNC_OK being valid.
9		Either PREAM_OK or NODE_OK being valid.
10		Either PREAM_OK or SYNC_OK or NODE_OK being valid
11	Once the detection conditions are met in T1, it will switch to T2 and once SYNC is detected in T2, it will exit T2 and hands over control to MCU; otherwise, it will exit RX after T2 finishes timing	RSSI_PJD_VALID being valid
12		PREAM_OK being valid
13		Both RSSI_PJD_VALID and PREAM_OK being valid at the same time.

Some of the above 14 SLP modes use RSSI_VLD as the triggering condition. It is an innovation of CMT2310A product to use RSSI_PJD_VALID generated by PJD to assist ultra-low power reception, which has remarkable effect and is recommended for users. Option 3 combines RSSI_PJD_VALID and PREAM_OK generated by PJD to make judgment conditions, which increases reliability without increasing too much time

If option 3 is used, assuming the transmitter sends a preamble long enough, T1 can be set by the following method:

Reserve 8 symbols for transmitter to do AFC, set PJD jump number as 6 and long preamble length as 4 or 8 symbols, then add a time of 6 - 8 symbols to detect the valid extension condition of RSSI_PJD_VALID& PREAM_OK. Therefore, the total T1 time is 14 – 16 symbols. If the crystal frequency deviation between TX and RX is not large, for example, far smaller than the configured deviation, then the front 8 symbols for AFC can be relatively reduced, which requires practical test of users.

2.4.1 SLP Mode 0

When the SLP mode is set to 0, the receiving time is equal to RX T1 without any extension processing. As can be seen from the diagram below, MCU is only responsible for sending go_rx and go_sleep commands to switch the state. The CMT2310A will automatically wake up to RX and exit RX with corresponding interruption.

The time such as RX and SLEEP given in the figure are only for illustration and have no special significance

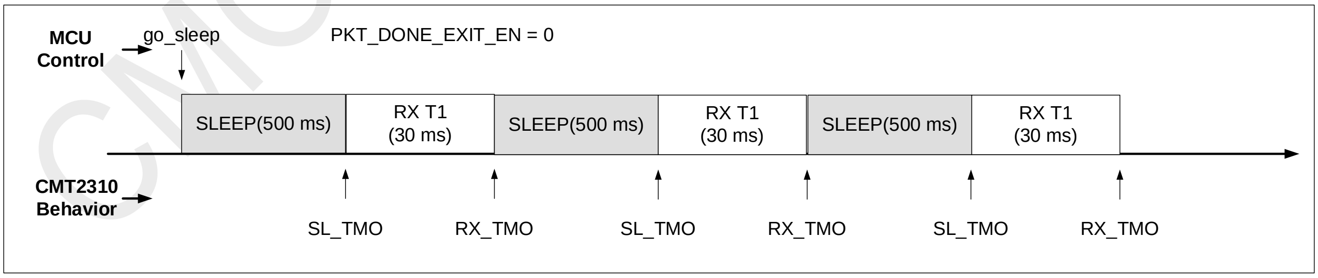


Figure 16. Using go_sleep under SLP Mode 0

CMOSTEK Confidential

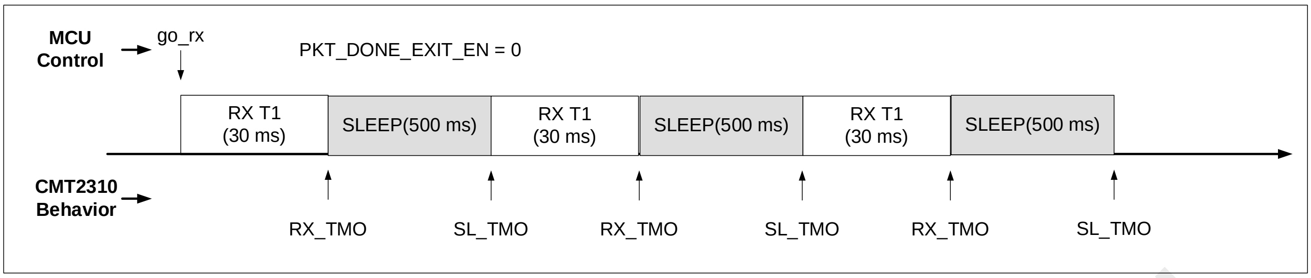


Figure 17. Using go_rx under SLP Mode 0

2.4.2 SLP Mode 1-3

When the mode is set to 1-3, once the detection conditions are met in RX T1, RX T1 stops timing and the chip stays in RX, and the control is handed over to MCU. Otherwise, RX T1 will exit at the end of the timer. The three different conditions are as follows

1. The detection condition is RSSI_VLD being valid.
2. The detection condition is PREAM_OK being valid.
3. The detection condition is that both RSSI_VLD and PREAM_OK are valid at the same time.

Taking option 2 as an example, the sequence diagram of TX, MCU and RX working together is shown as followed:

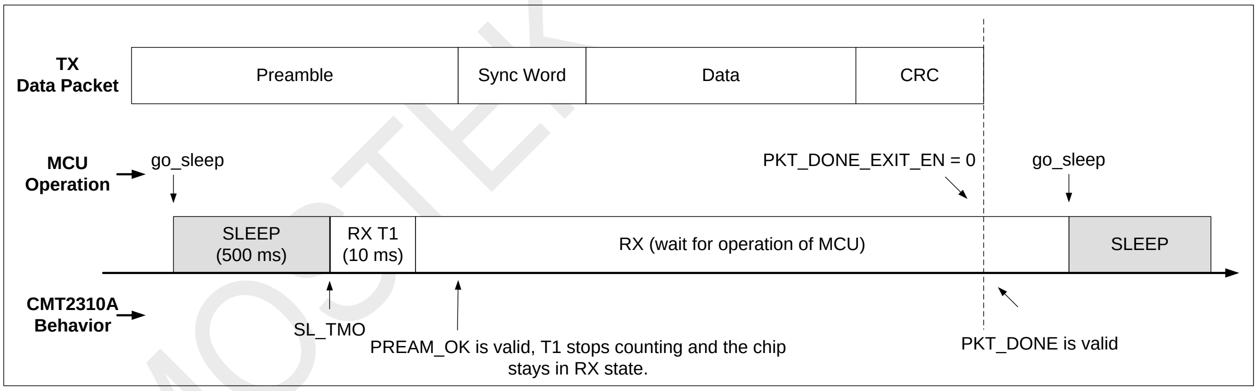


Figure 18. PKT_DOEN_EXIT_EN = 0 in SLP Mode 2

CMOSTEK Confidential

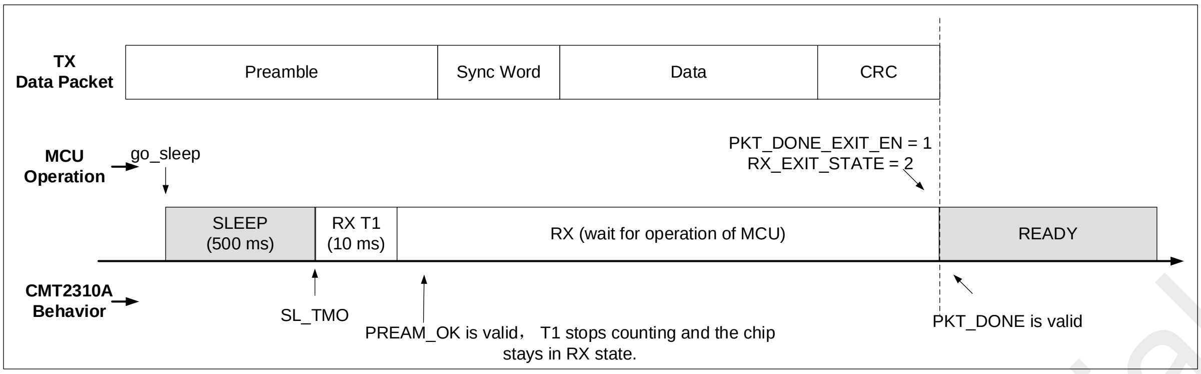


图 19. PKT_DOEN_EXIT_EN = 1 in SLP Mode 2

It should be noted that the preset time of RX T1 is 10 ms, however once the condition is met, it will stop timing, and will reset and re-timing at the next time when enter into RX. When the user needs the chip to return to the state specified by RX_EXIT_STATE after successfully receiving Packet in SLP mode, PKT_DONE_EXIT_EN should be set to 1.

The following scenarios are suitable from pattern 1 to 3:

In some applications, the TX content varies greatly, that is, the transmitted content, packet length and packet numbers are variable each time. When the detection conditions are met and RX T1 stops, the switching right of RX is handed back to MCU, and MCU decides the receiving time by itself.

The following shows comparisons among 3 kinds of conditions:

For option 1, the detection condition RSSI_PJD_VLD is the output of channel listening. As mentioned above, channel listening, assisted by PJD, can make a very fast and reliable judgement of whether the incoming signal is useful or not. More importantly, it does not limit by data format and a very long Preamble transmitted, etc. Therefore, the benefit of option 1 is that when the RX T1 window is set to very small (such as 5-8 symbols), it is still very reliable for detection judgement.

For option 2, the detection condition PREAM_OK is the interrupt generated from Packet, which is a traditional detection condition. Generally，at least 2 bytes Preamble length can ensure reliable detection, which makes it easier to be comprehended. However, it’s disadvantage is that preamble must be contained in the packet format and the length of preamble transmitted at TX side must be long enough (covering double RX T1 time and 1 SLEEP time). What’s more, RC T1 has to cover Preamble with 2 or more bytes.

For option 3, The detection condition is that both of them are valid at the same time, which means that the detection condition is relatively harsh and extremely difficult to be triggered by mistake, with very high reliability. The minimum value of RX_PREAM_SIZE of CMT2310 can be set to 4 symbols. If combined with RSSI_PJD_VLD signal, the time of RX T1 will not be increased and more reliable detection effect can be achieved.

2.4.3 SLP Mode 4

When it is set as SLP mode 4, once it detects RSSI_PJD_VLD valid within RX T1, it will stops counting and stays in RX

CMOSTEK Confidential

until RSSI_PJD_VLD becoming invalid, then it exits RX automatically; if it does not detect RSSI_PJD_VLD valid within RX T1, it will exit RX when counting ends.

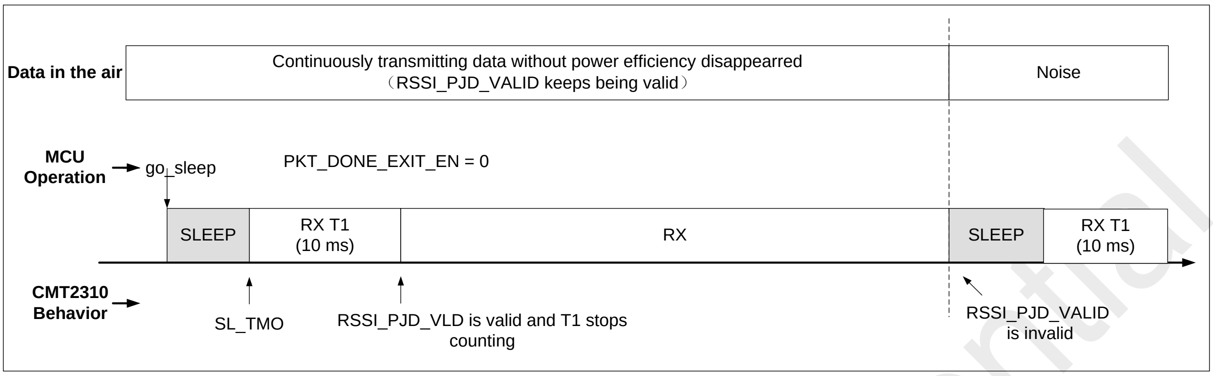

Data in the air
Continuously transmitting data without power efficiency disappearred
（RSSI_PJD_VALID keeps being valid）
Noise
MCU Operation
go_sleep
PKT_DONE_EXIT_EN = 0
SLEEP
RX T1 (10 ms)
RX
SLEEP
RX T1 (10 ms)
CMT2310 Behavior
SL_TMO
RSSI_PJD_VLD is valid and T1 stops counting
RSSI_PJD_VALID is invalid

Figure 20. SLP Mode 4，RSSI_PJD_VALID being valid

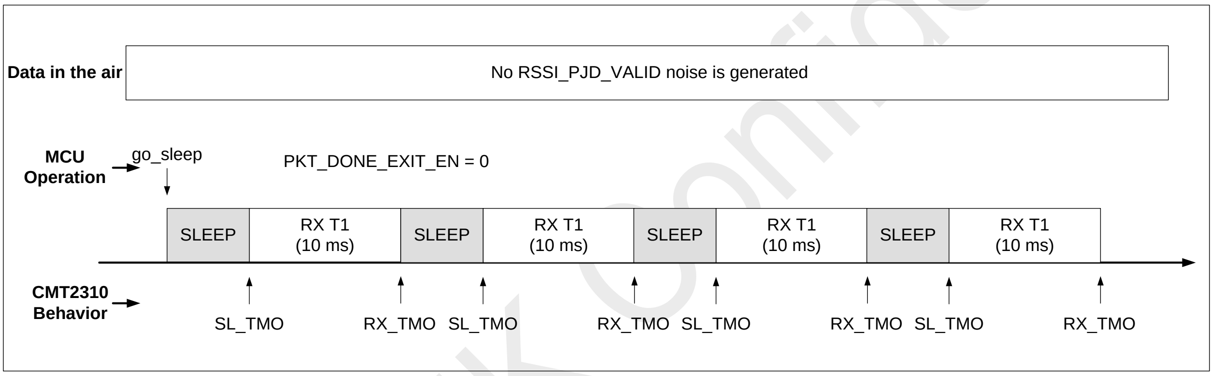


Figure 21. SLP Mode 4，RSSI_PJD_VALID being invalid

CMOSTEK Confidential

2.4.4 SLP Mode 5-10

When SLP mode is set as 5 ~ 10, once it met the detection condition within RX T1, it switches to RX T2 and exits RX automatically when RX T2 counting ends. Otherwise it will exit RX when counting ends. The 6 different conditions are as follows.

1.The detection condition is RSSI_PJD_VLD being valid.

2.The detection condition is PREAM_OK being valid.

3.The detection condition is that both RSSI_PJD_VLD and PREAM_OK being valid at the same time.

4.The detection condition is either PREAM_OK or SYNC_OK being valid.

5.The detection condition is either PREAM_OK or NODE_OK being valid.

6.The detection condition is any one of PREAM_OK , SYNC_OK or NODE_OK being valid.

Taking option 6 as an example, sequence diagram for co-operating among TX, MCU and RX is shown in the figure below.

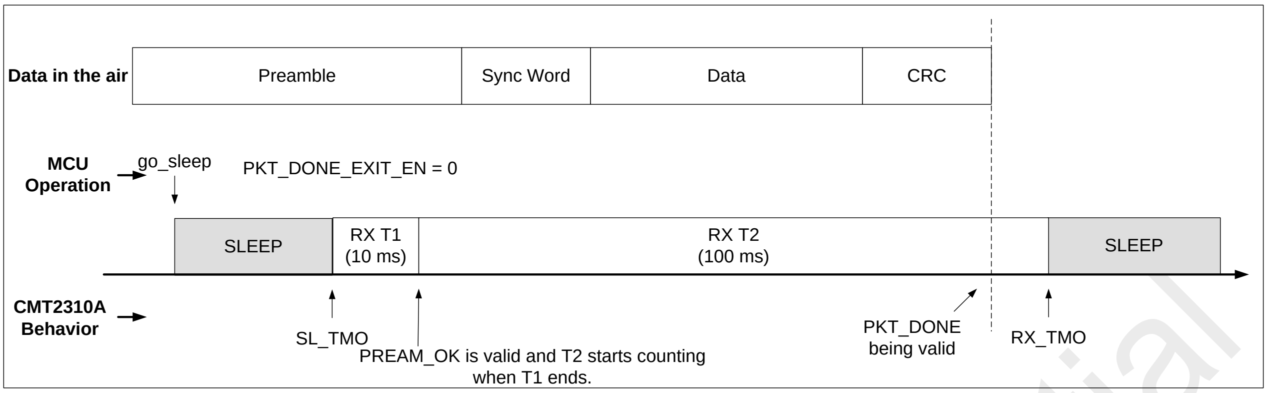


Figure 22. SLP Mode 6, T1 detects condition for switching to T2

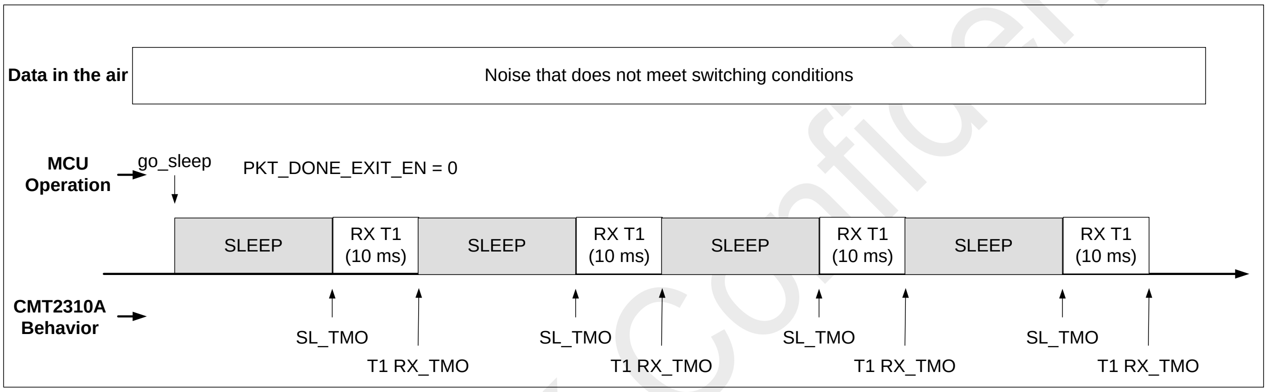


Figure 23. SLP Mode 6，T1 cannot meet the condition of switching to T2

CMOSTEK Confidential

Users need to set RX T2 long enough to receive all the required content.

Mode 5 - 10 fit for the scenarios below:

Comparing with mode 1 -3, mode 5 -10 vary in the presence of RX T2. When the chip switches to RX T2, neither MCU's participation for control nor timeout made by MCU is required; the chip will exit RX automatically when RX T2 overtime under the condition of PKT_DONE_EXIT_EN disabled. Therefore, mode 5- 10 are suitable for applications where the data length is similar each time,which makes the time of RX T2 easier to set.

The practical benefits of Mode 8 - 10:

In some applications, the low-power transceiver scheme does not comply with the calculation principles described above for the reason that users may try to minimize the power consumption of TX/RX. For example, the RX time is set relatively short, so that it can only capture detection conditions sent from TX in some probability. Setting the detection conditions to be valid for any one of two or three conditions will increase the probability of successful capture, but at the same time reduce the reliability. If verified in market that RSSI_PJD_VLD assisted by PJD can work reliably, then option 8 - 10 are basically of no practical use.

2.4.5 SLP Mode 11-13

When it is set as SLP mode 11 -13, once detection conditions are met within RX T1, it switches to RX T2. When SYNC_OK is detected to be valid within RX T2, then RX T2 stops counting and the chip will stay in RX and hand over control to MCU; Therefore, if it does not detect conditions valid within RX T1 or RX T2, then it exits RX when RX T1 or RX T2 ends. The 3 different conditions are as follows.

1. The detection condition is RSSI_PJD_VLD being valid.
2. The detection condition is PREAM_OK being valid.
3. The detection condition is both RSSI_PJD_VLD and PREAM_OK being valid at the same time.

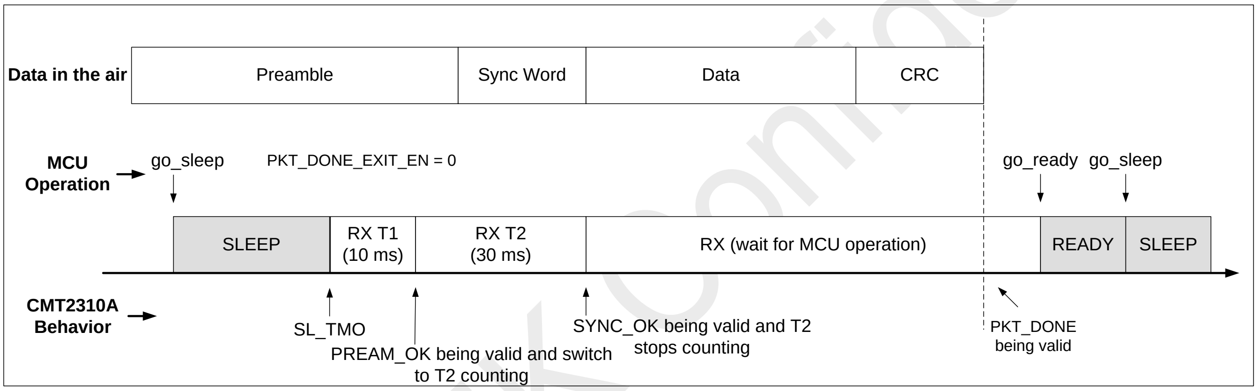


Figure 24. SLP Mode 12

Mode 11 - 13 fit for the scenarios below:

All three modes are available for data packets with applications of Preamble and Sync Word.The advantage of these three modes is that if there are different transmission data in the same channel, only the detection of Preamble or RSSI cannot ensure that the received data is the desired ones. Here it comes the second level detection condition of Sync Word fixing, which makes it reliable. If the secondary detection condition is not satisfied, it will exit after RX T2, thus saving power from false-triggering.

CMOSTEK Confidential

2.5 TX Duty Cycle Mode Description

TX Duty Cycle mode is used to continuously transmit packets so that the chip is in the automatic TX - SLEEP sending process. It is divided into 2 modes.

The number of automatic sending number in non-persistent mode has an upper limit, which is configured by TX_DC_TIMES. TX_DC_DONE_TIMES reflects the transmitting times that have been completed. TX_DC_DONE_FLAG indicates the transmitting numbers of TX_DC_TIMES has been configured in non-persistent mode. The chip will then return to the READY state for MCU operation.

Persistent mode is applied when TX_DC_PERSIST_EN is set to 1. In persistent mode, there is no limit on the automatically transmitting numbers, the chip will automatically transmitting under TX -- SLEEP mode. By setting API_STOP to 1, the chip will exit and return to READY at the next state switch. Sequence diagrams for the two modes are shown below.

2.5.1 Non-persistent Mode

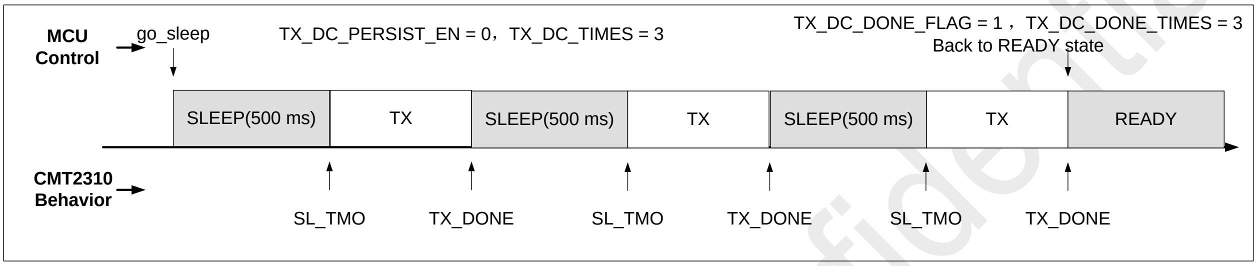


Figure 25. TX_DC in Non-persistent Mode

2.5.2 Persistent Mode

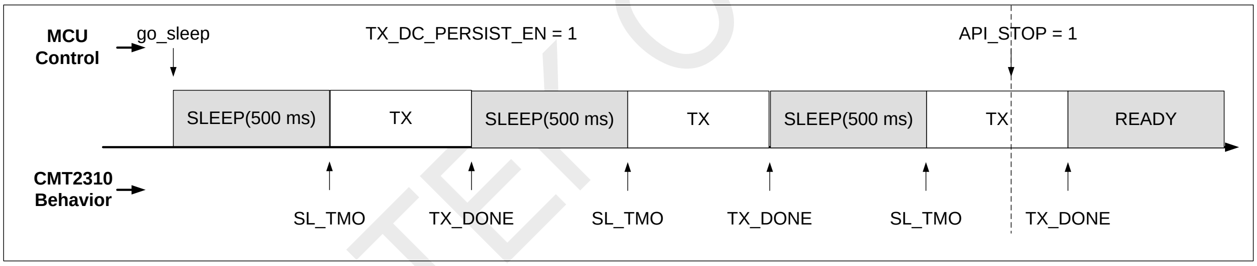


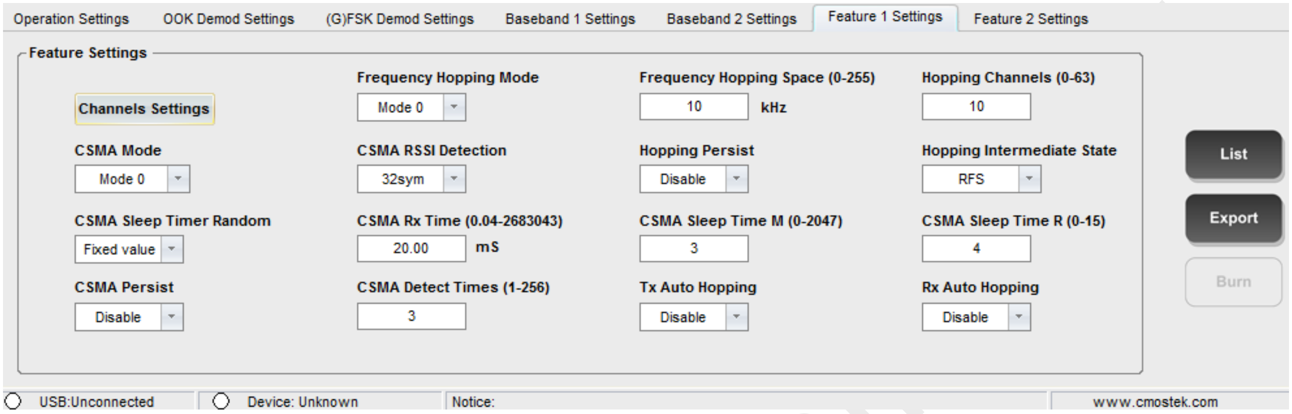
Figure 26. TX_DC in Persistent Mode

CMOSTEK Confidential

3. RX Auto Hop

3.1 Related Register in RX Auto Hop Mode

The corresponding RFPDK interface and parameters are shown as followed:



Feature Settings

Frequency Hopping Mode: Mode 0

Frequency Hopping Space (0-255): 10 kHz

Hopping Channels (0-63): 10

CSMA Mode: Mode 0

CSMA RSSI Detection: 32sym

Hopping Persist: Disable

Hopping Intermediate State: RFS

CSMA Sleep Timer Random: Fixed value

CSMA Rx Time (0.04-2683043): 20.00 mS

CSMA Sleep Time M (0-2047): 3

CSMA Sleep Time R (0-15): 4

CSMA Persist: Disable

CSMA Detect Times (1-256): 3

Tx Auto Hopping: Disable

Rx Auto Hopping: Disable

Figure 27. RX Auto Hop Related Parameters on RFPDK Interface

CMOSTEK.com

Table 18. RX Auto Hop Related Parameters

Register Bit RFPDK Parameter	Register Bit
Frequency Hopping Mode	RX_HOP_SLP_MODE<2:0>
Frequency Hopping Space	FREQ_SPACE<7:0>
Hopping Channels	FREQ_TIMES<7:0>
Hopping Persist	RX_HOP_PERSIST
Hopping Intermediate State	FREQ_SW_STATE
Rx Auto Hopping	RX_AUTO_HOP_EN

The contents and description of the registers are detailed in the table below.

Table 19. Related Registers in Configuration Area

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_11 (0x0B)	7:0	RW	FREQ_DONE_TIMES	rx_auto_hop refers to the completed hopping times. When rx_auto_hop enabled and complete one hopping it will add 1 automatically. Users can achieve the required hopping times though configuration of the value and the freq_times；When users configure the value over 64，the chip will thought the configuration as 0.

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_13 (0x0D)	7:0	RW	FREQ_TIMES	Configure hopping table of rx_auto_hop. If the maximum depth of the hopping table is 64 and the configuration is over 64, it is thought to be configured as the deepest depth 64.
CTL_REG_12 (0x0C)	7:0	RW	FREQ_SPACE	Channel interval of automatic frequency hopping
CTL_REG_22 (0x16)	6	RW	RX_HOP_PERSIST	When the bit works on rx_auto_hop after skipping the full table, the bit determines whether to continue the hopping scan from the header or return to the corresponding state by rx_exit_state. 1：Continue the hopping scan from the start 0：Back to the rx_exit_state corresponding state
	5	RW	FREQ_SW_STATE	The bit works on rx_auto_hop after the current channel window, the bit determines whether to configure the next hop channel in the READY or RFS state, and then enters the RX to scan the signal. 1：Back to the RFS channel to configure the next channel 0：Back to the READY channel to configure the next channel
CTL_REG_37 (0x25)	7:0	R	REQ_CHANL_LV	In the rx_auto_hop mode, the currently used channel value will be provided to the user for further operation.
CTL_REG_97 (0x61)	1	RW	RX_AUTO_HOP_EN	Rx Auto Hop functional enabling bit 0：disable 1：enable
CTL_REG_98 (0x62)	2:0	RW	RX_HOP_SLP_MODE <2:0>	Rx Auto Hop Configuration Valid range: 0-6

CMOSTEK Confidential

3.2 Functional Using Description

There are 7 modes in RX_Auto_Hop, which is configured through RX_HOP_SLP_MODE. In RX Auto Hop, FREQ_TIMES<7:0> is used to configure the size of frequency hopping table, which contains a space up to 64 bytes to save configured signal channels for users.

The chip enters a signal channel to receive. If no successful events are detected in the corresponding mode, then the chip will return to either READY state or RFS state according to FREQ_SW_STATE. After then it configures the next channel value in the frequency hopping table automatically and returns to RX to receive. If automatic switching state is READY, PLL frequency will be calibrated before it reenters RX, thus it will cost calibration time. If the automatic switching state is RFS, PLL frequency will not be calibrated, thus the switching time will be shorter. If the channel space between any of two frequency points is more than 2 MHz, it is suggested that FREQ_SW_STATE is configured as READY since re-calibration is needed in this case.

If successful events are detected, auto-hopping will stop and it waits for the external MCU to have further processing or it will switch to the preconfigured state. If no successful events are detected when the entire table are hopped, providing RX_HOP_PERSIST is set as 1, the chip will continue hopping from the beginning of the hopping table; providing RX_HOP_PERSIST is set as 0, the chip returns to RX_EXIT_STATE state to accomplish the auto-hopping RX operation.

The target frequency point of frequency hopping is calculated as follows:

FREQ = 1K x FREQ_SPACE<7:0> x FREQ_CHANL<7:0>

FREQ_SPACE<7:0> is an independent register and FREQ_CHANL<7:0> is configured through 64 registers with each register storing a channel value.

In RX_Auto_Hop mode, register REQ_CHANL_LV is available to tell users which channel value is currently used. Furthermore, register FREQ_DONE_TIMES is used to read the current completed numbers of hopping.

In RX_Auto_Hop mode, registers RX T1 TIMER and RX T2 TIMER are used to involve in automatic control, thus users must set TIMER_RX_EN to 1. RX T1 TIMER is configured through TIMER_M_RX_T1 and TIMER_R_RX_T1 and the actual timing time can be calculated according to the formula; RX T2 TIMER is configured through TIMER_M_RX_T2 and TIMER_R_RX_T2 and the actual timing time can be calculated according to the formula;

CCA_INT_SEL is used to determine whether RSSI_PJD_VLD is judged by RSSI or PJD. RX_EXIT_STATE is used to judge which state will be returned when it exits RX. PKT_DONE_EXIT_EN is used to judge whether to stay in current state or return to the state configured in RX_EXIT_STATE when PKT_DONE is received.

When the configurations are accomplished, users set RX_AUTO_HOP_EN to 1 to enable the function, and then send command go_rx to enter RX_Auto_Hop mode. During the automatic operating, if events to trigger stop event, the chip will exit the automatic operating and stay in a certain state. If users would like to stop RX_Auto_Hop function on their accord, they can set API_STOP to 1. When the chip detects this bit, then it will set API_DONE_FLAG to 1 upon next state switching and then keep the current configuration and exit RX_Auto_Hop, and then return to READY state.

When RX Auto Hop mode stops. if RX_AUTO_HOP_EN keeps as 1, then when users send command go_rx the next time, the chip will reenter RX Auto Hop mode and start hopping from the frequency point next to the stopped frequency point last time. Users can clear register FREQ_DONE_TIMES right before sending command go_rx.

RX Auto Hop consists of 7 operating modes as listed in the table below

CMOSTEK Confidential

Table 20. RX Auto Hop Operating Mode

No.	RX Extension Method	RX Extension Condition
0	If it is configured as 0, there's no extension. If PKT_DONE is detected within T1 and PKT_DONE_EXIT_EN = 1, then it exits RX Auto-Hop and switches to the state defined in RX_EXIT_STATE; otherwise，it exits RX when T1 time ends and	None

No.	RX Extension Method	RX Extension Condition
	returns to the state defined in FREQ_SW_STATE and continues to hop to the next frequency point.	
1	Once detection conditions are met within T1, it exits T1 and hands over control to MCU (keep current state); Otherwise，if the detection conditions are not met within T1, it exits RX when T1 time ends and returns to the state defined in FREQ_SW_STATE and then continue to hop to the next frequency point.	RSSI_PJD_VLD being valid
2		PREAM_OK being valid
3		RSSI_PJD_VLD and PREAM_OK being valid at the same time
4	Once the detection conditions are met within T1, it switches to T2; Otherwise，if the detection conditions are not met within T1, it exits RX and returns to the state defined in FREQ_SW_STATE and then continues to hop to the next frequency point. Once SYNC is detected within T2, it exits T2 and hands over control to MCU (keep the state); otherwise it exits T2 when T2 time ends and returns to the state defined in FREQ_SW_STATE and then continues to hop to the next frequency point.	RSSI_PJD_VLD being valid
5		PREAM_OK being valid
6		RSSI_PJD_VLD and PREAM_OK being valid at the same time.

CMOSTEK Confidential

3.3 Sequence Diagram Description

3.3.1 RX Auto Hop Table Hopping Method

When detection is not successful after the entire table is hopped over, there are 2 processing methods. If RX_HOP_PERSIST_EN is 0, the chip will exit the automatic control of RX Auto Hop and return to RX_EXIT_STATE state; if RX_HOP_PERSIST_EN is 1, the chip will hop again starting from the beginning of the table to detect signal.

1. RX_HOP_PERSIST_EN = 0

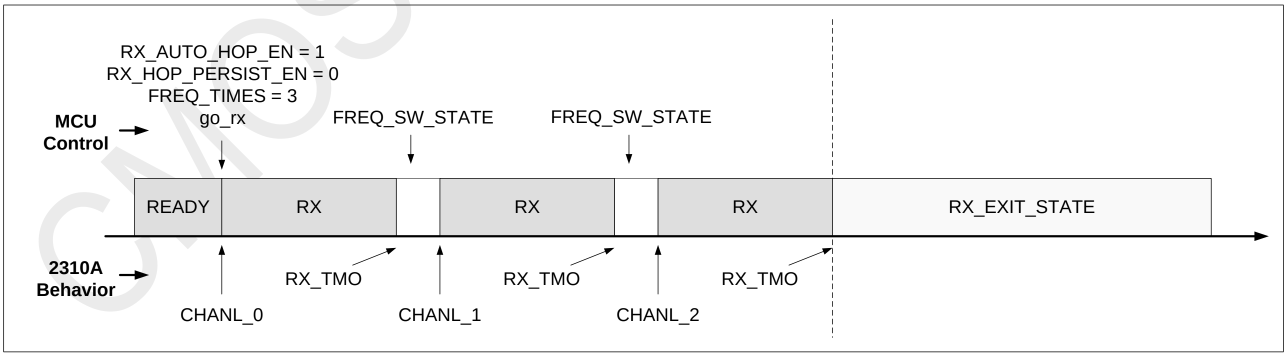


Figure 28. RX_HOP_PERSIST_EN = 0 Exit after Completing Table Hopping

2. RX_HOP_PERSIST_EN = 1

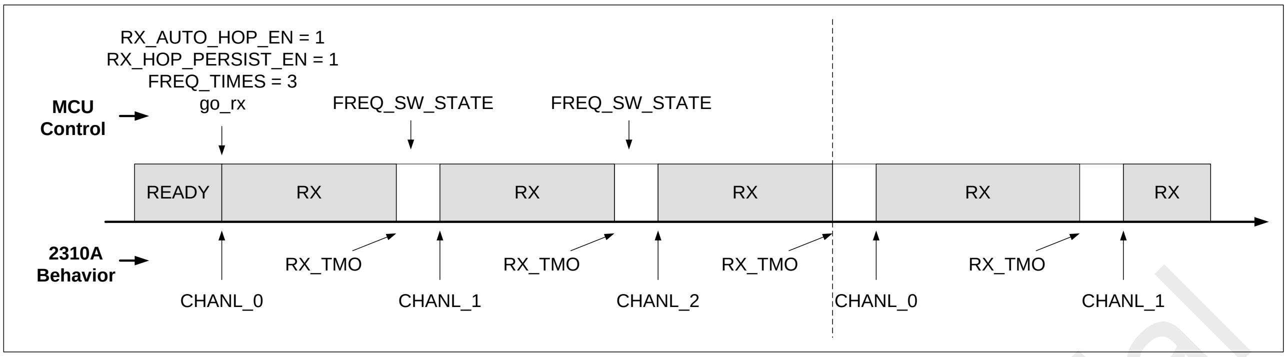


Figure 29. RX_HOP_PERSIST_EN = 1 Hopping in Cycle

3.3.2 RX Auto Hop Mode 0

When mode 0 is set, the receiving time is equal to RX T1 with no extension processing. If PKT_DONE is detected before T1 timeout and meanwhile PKT_DONE_EXIT_EN = 1, it then exits RX Auto Hop and switches to the state defined in RX_EXIT_STATE. Here RX Auto Hop is stopped; otherwise，it exits RX when T1 time ends and returns to the state defined in FREQ_SW_STATE and continues to hop to the next frequency point.

As shown in the diagram below, MCU only responsible for sending go_rx command to switch state and the CMT2310A will have automatic frequency points switching and state switching, thus generate corresponding interrupts.

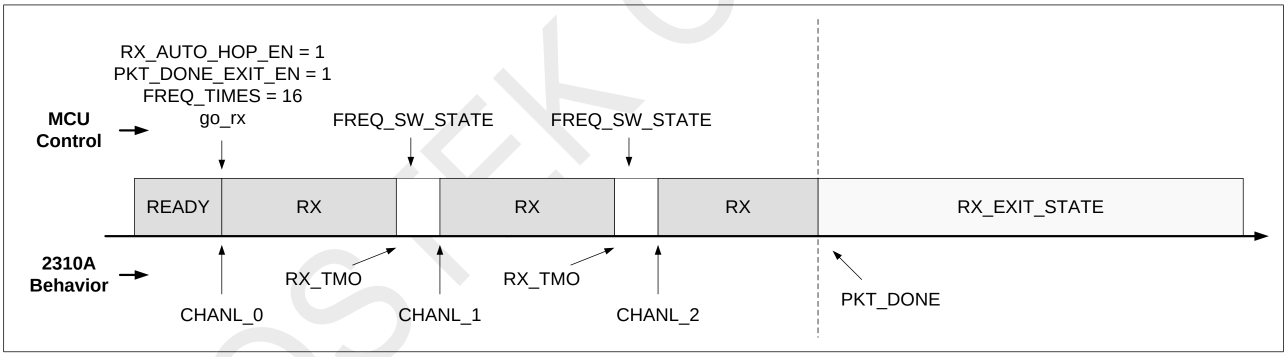


Figure 30. RX Auto Hop Mode 0

3.3.3 RX Auto Hop Mode 1-3

If mode 1- 3 are set, once detection conditions are met within RX T1, it exits RX T1; the chip stays in RX and hands over control to MCU or switch to the state defined in RX_EXIT_STATE automatically. Here Rx Auto Hop is stopped; otherwise upon RX T1 timeout it exits RX and switches to the state defined in FREQ_SW_STATE and continues to hop to the next frequency point. The different detection conditions are as follows.

1. The detection condition is RSSI_VLD being valid.

2. The detection condition is PREAM_OK being valid.

CMOSTEK Confidential

3. The detection condition is both RSSI_PJD_VLD and PREAM_OK being valid at the same time.

Taking option 2 as an example, the sequence diagram of TX, MCU and RX successfully receiving packets in a frequency point is presented below. In diagram-1, it hands over control to MCU when receiving data packet successfully. In diagram-2, it switches to the state defined in RX_EXIT_STATE automatically.

1. PKT_DONE_EXIT_EN = 0

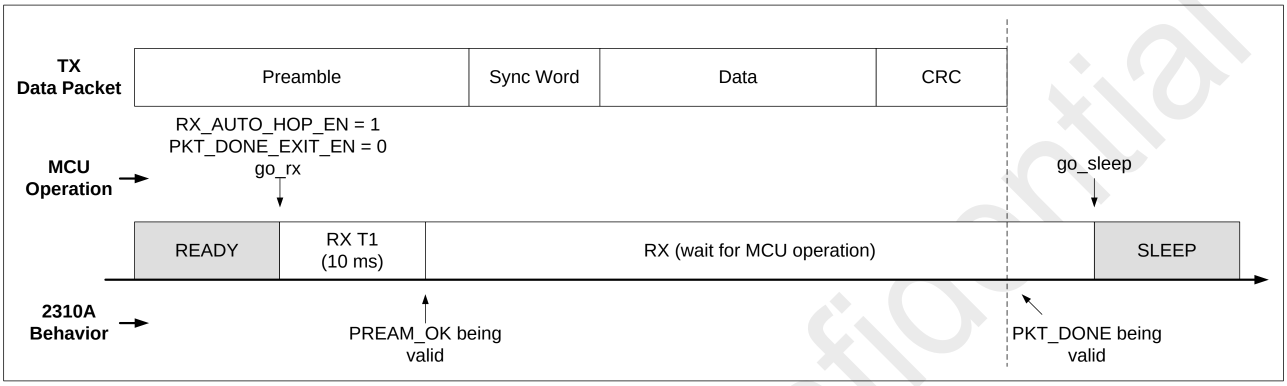


Figure 31. RX Auto Hop Mode 2，PKT_DONE_EXIT_EN = 0

2. PKT_DONE_EXIT_EN = 1

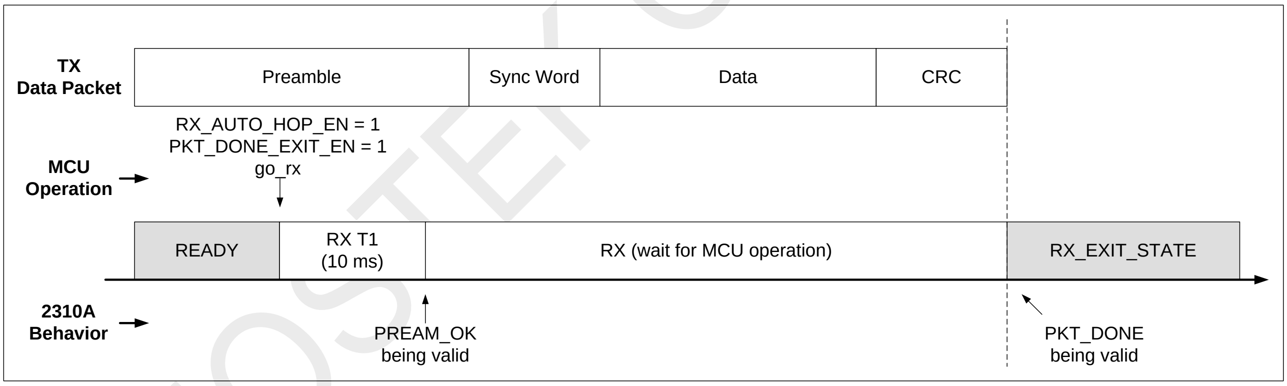


Figure 32. RX Auto Hop Mode 2，PKT_DONE_EXIT_EN = 1

3.3.4 RX Auto Hop Mode 4-6

If mode 4- 6 are set, once detection conditions are met within RX T1, it switches to RX T2; If SYNC_PASS is detected valid within RX T2, it hands over control to MCU or switches to RX_EXIT_STATE and Rx Auto Hop is stopped at this time; otherwise, it exits RX upon RX T2 timeout and continues to hop to the next frequency point. If the detection conditions are not met within RX T1, it exits Rx when timer ends and continues to hop to the next frequency point. The three different conditions are as follows.

1. The detection condition is RSSI_VLD being valid.

CMOSTEK Confidential

2. The detection condition is PREAM_OK being valid.
3. The detection condition is both RSSI_PJD_VLD and PREAM_OK being valid at the same time.

Taking option 5 as an example, the sequence diagram of TX,MCU and RX working together is shown below:

1. PKT_DONE_EXIT_EN = 0

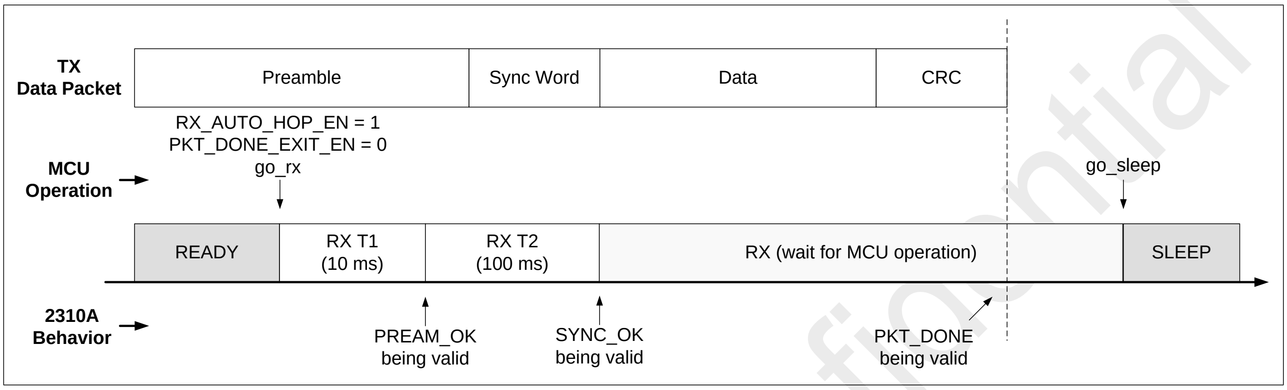


Figure 33. RX Auto Hop Mode 5，PKT_DONE_EXIT_EN = 0

2. PKT_DONE_EXIT_EN = 1

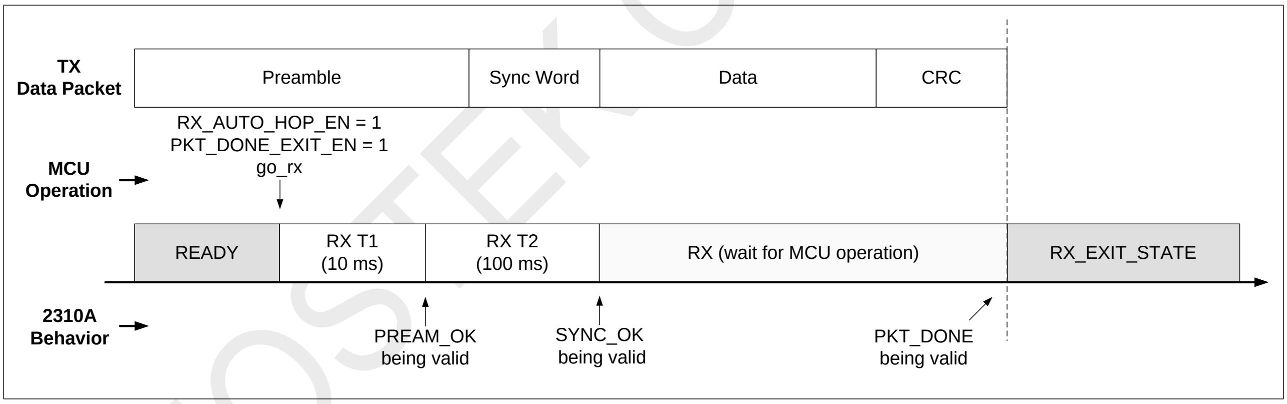


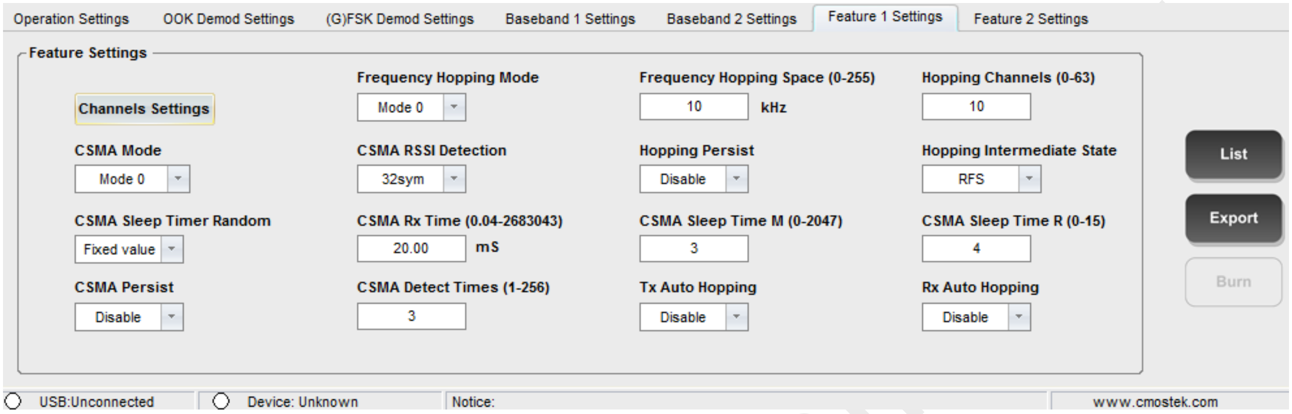
Figure 34. RX Auto Hop Mode 5，PKT_DONE_EXIT_EN = 1

CMOSTEK Confidential

4. TX Auto Hop

4.1 Related Register of TX Auto Hop

The corresponding RFPDK interface and parameters are shown as followed:



Operation Settings | OOK Demod Settings | (G)FSK Demod Settings | Baseband 1 Settings | Baseband 2 Settings | Feature 1 Settings | Feature 2 Settings

Feature Settings

Channels Settings

Frequency Hopping Mode: Mode 0

Frequency Hopping Space (0-255): 10 kHz

Hopping Channels (0-63): 10

CSMA Mode: Mode 0

CSMA RSSI Detection: 32sym

Hopping Persist: Disable

Hopping Intermediate State: RFS

CSMA Sleep Timer Random: Fixed value

CSMA Rx Time (0.04-2683043): 20.00 mS

CSMA Sleep Time M (0-2047): 3

CSMA Sleep Time R (0-15): 4

CSMA Persist: Disable

CSMA Detect Times (1-256): 3

Tx Auto Hopping: Disable

Rx Auto Hopping: Disable

Figure 35. Related Parameters of RFPDK Interface in TX Auto Hop

CMOSTEK Con

Table 21. TX Auto Hop Related Parameters

Register Bit RFPDK Parameter	Register Bit
Frequency Hopping Space	FREQ_SPACE<7:0>
Hopping Channels	FREQ_TIMES<7:0>
Hopping Persist	RX_HOP_PERSIST
Hopping Intermediate State	FREQ_SW_STATE
Tx Auto Hopping	TX_AUTO_HOP_EN

The contents and description of registers are shown in the following table.

Table 22. Register in the Configuration Area

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_11 (0x0B)	7:0	RW	FREQ_DONE_TIMES	tx_auto_hop refers to the completed hopping times. When tx_auto_hop enabled and complete one hopping it will add 1 automatically. Users can achieve the required hopping times though configuration of the value and the freq_times；When users configure the

Register Name	Bit	R/W	Bit Flag	Function Description
				value over 64， the chip will thought the configuration as 0.
CTL_REG_13 (0x0D)	7:0	RW	FREQ_TIMES	Configure hopping table of tx_auto_hop. If the maximum depth of the hopping table is 64 and the configuration is over 64, it is thought to be configured as the deepest depth 64.
CTL_REG_12 (0x0C)	7:0	RW	FREQ_SPACE	Channel interval of automatic frequency hopping
CTL_REG_22 (0x16)	6	RW	RX_HOP_PERSIST	When the bit works on tx_auto_hop after skipping the full table, the bit determines whether to continue the hopping scan from the header or return to the corresponding state by tx_exit_state. 1：Continue the hopping scan from the start 0：Back to the tx_exit_state corresponding state
	5	RW	FREQ_SW_STATE	The bit works on tx_auto_hop after the current channel window, the bit determines whether to configure the next hop channel in the READY or RFS state, and then enters the TX to scan the signal. 1：Back to the RFS channel to configure the next channel 0：Back to the READY channel to configure the next channel
CTL_REG_37 (0x25)	7:0	R	REQ_CHANL_LV	In the tx_auto_hop mode, the currently using channel value will be provided to the user for further operation.
CTL_REG_96 (0x60)	1	RW	TX_AUTO_HOP_EN	Tx Auto Hop Functional Enabling Bit 0：disable 1：enable

CMOSTEK Confidential

4.2 Functional Using Description

FREQ_TIMES<7:0> in TX Auto Hop is used for configuring the size of the frequency table, which contains a maximum of 64 bytes for saving the required configured signal channel. The process of frequency hopping is as follows.

The chip enters a signal channel to transmit. After sending successfully, the chip returns to the READY or TFS state according to FREQ_SW_STATE and automatically configure the channel value of the next byte in the frequency hopping table, and then re-enters the TX state for transmitting. If automatic switching state is READY, PLL frequency will be calibrated before it reenters TX, thus it will cost calibration time. If the automatic switching state is TFS, PLL frequency will not be calibrated, thus the switching time will be shorter. If the channel space between any of two frequency points is more than 2 MHz, it is suggested that

FREQ_SW_STATE is configured as READY since re-calibration is needed in this case.

When the full hopping table finished, if the RX_HOP_PERSIST is configured as 1, the chip will start hopping from the begin, if RX_HOP_PERSIST is configured as 0, the chip will back to TX_EXIT_STATE and stops the auto-hopping transmission.

The target frequency point of frequency hopping is calculated as follow:

FREQ = 1K x FREQ_SPACE<7:0> x FREQ_CHANL<7:0>

FREQ_SPACE<7:0> is an independent register and FREQ_CHANL<7:0> is configured through 64 registers with each register storing a channel value.

In TX_Auto_Hop mode, register REQ_CHANL_LV is available to tell users which channel value is currently used. Furthermore, register FREQ_DONE_TIMES is used to read the current completed numbers of hopping.

When the configurations are accomplished, users set TX_AUTO_HOP_EN to 1 to enable the function, and then send go_rx command to enter TX_Auto_Hop mode. If users would like to stop TX_Auto_Hop function on their accord, they can set API_STOP to 1. When the chip detects this bit, it will set API_DONE_FLAG to 1 upon next state switching and then keep the current configuration and exit TX_Auto_Hop, and then return to READY state.

4.3 Sequence Diagram Description

- **HOP_PERSIST_EN = 0**

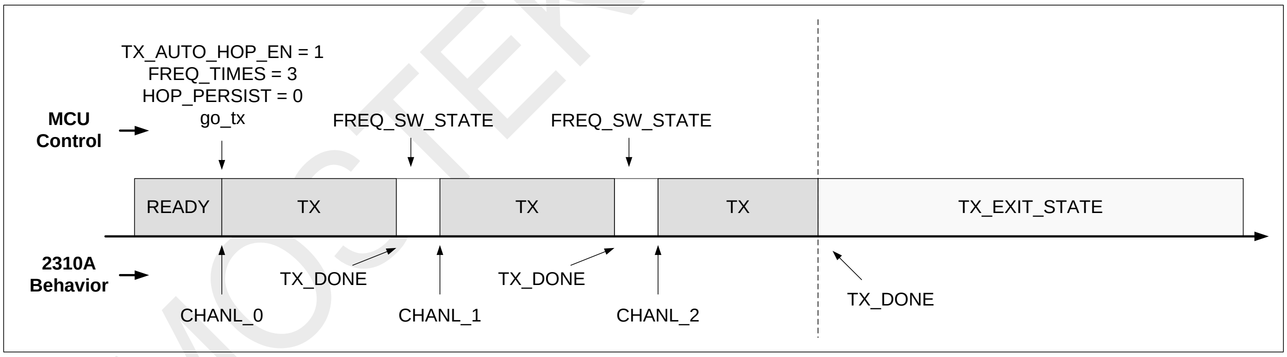


Figure 36. TX Auto Hop，HOP_PERSIST_EN = 0

- **HOP_PERSIST_EN = 1**

CMOSTEK Confidential

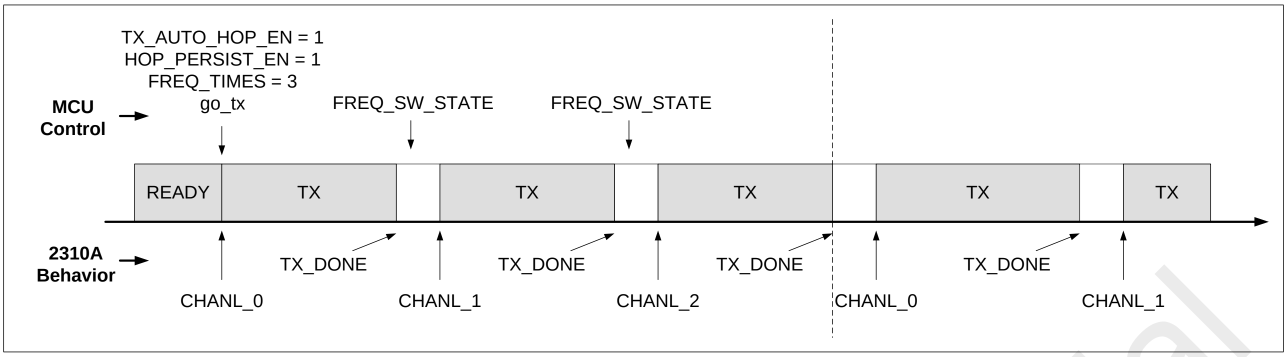
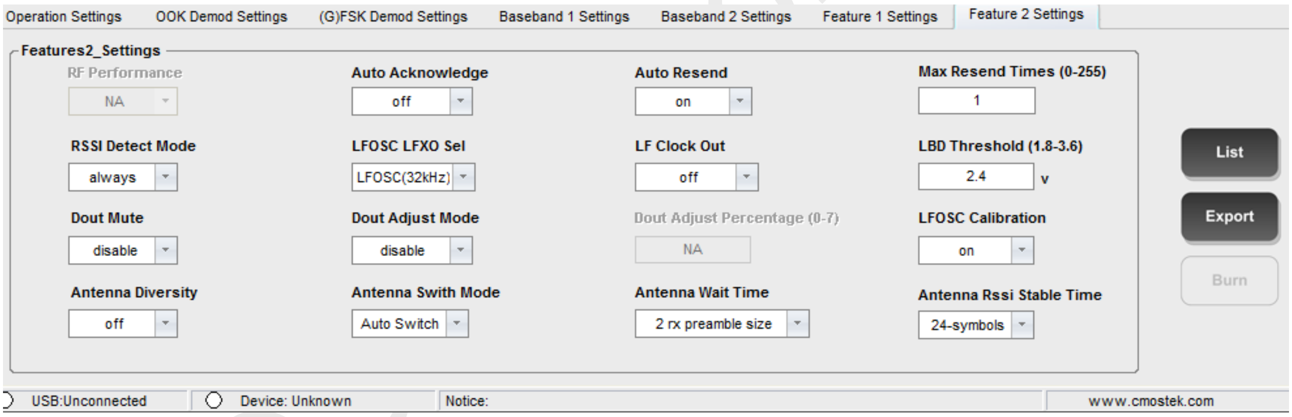


Figure 37. TX Auto Hop，HOP_PERSIST_EN = 1

5. TX Auto Resend

5.1 Related Registers of Auto Resend Mode

The corresponding RFPDK interface and parameters are shown as follow.



RF Performance: NA
Auto Acknowledge: off
Auto Resend: on
Max Resend Times (0-255): 1
RSSI Detect Mode: always
LFOSC LFXO Sel: LFOSC(32kHz)
LF Clock Out: off
LBD Threshold (1.8-3.6): 2.4 v
Dout Mute: disable
Dout Adjust Mode: disable
Dout Adjust Percentage (0-7): NA
LFOSC Calibration: on
Antenna Diversity: off
Antenna Swith Mode: Auto Switch
Antenna Wait Time: 2 rx preamble size
Antenna Rssi Stable Time: 24-symbols

Figure 38. Related Parameters on RFPDK in TX Auto Resend

Table 23. Related Parameters in TX Auto Resend

Register Bit RFPDK Parameter	Register Bit
Auto Resend	TX_ACK_EN
Max Resend Times	TX_RESEND_TIMES

The content and description of related registers are described in the table below.

Table 24. Related Registers in Configuration Area

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_96 (0x60)	2	RW	TX_ACK_EN	Enabling RESEND Function under TX Mode 0：disable 1：enable
CTL_REG_113 (0x71)	7:0	RW	TX_RS_TIMES<7:0>	The maximum resend times under TX Resend mode.
CTL_REG_23 (0x17)	3	RW	ACK_RECV_FAILED_EN	ACK_RECV_FAILED_FLAG and ACK_RECV_FAILED interrupt enabling configuration bit. After enabling, if the corresponding event of ACK_RECV_FAILED_FLAG is valid, ACK_RECV_FAILED_FLAG will be set as 1 and generate ACK_RECV_FAILED interruption. 0：disable 1：enable
	2	RW	TX_RESEND_DONE_EN	TX_RESEND_DONE_FLAG and TX_RESEND_DONE interrupt enabling configuration bit. After enabling, it reaches the maximum numbers of unable received and error packet and TX_RESEND_DONE_FLAG will be set as 1，thus generate TX_RESEND_DONE interruption. 0：disable 1：enable
	1	RW	NACK_RECV_EN	NACK_RECV_FLAG and NACK_RECV interrupt enabling configuration bit. After enabling, if the corresponding event of NACK_RECV_FLAG is valid，NACK_RECV_FLAG will be set as 1 thus generate NACK_RECV interruption. 0：disable 1：enable
	0	RW	SEQ_MATCH_EN	SEQ_MATCH_FLAG and SEQ_MATCH interrupt enabling configuration bit. After enabling, if the corresponding event of SEQ_MATCH_FLAG is valid, SEQ_MATCH_FLAG will be set as 1 and generate SEQ_MATCH interruption. 0：disable 1：enable

CMOSTEK Confidential

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_31 (0x1F)	3	W	ACK_RECV_FAILED_CLR	ACK_RECV_FAILED _FLAG clearing bit 0：NA 1：Clearing ACK_RECV_FAILED _FLAG
	2	W	TX_RESEND_DONE_CLR	TX_RESEND_DONE_FLAG clearing bit 0：NA 1：Clearing TX_RESEND_DONE_FLAG
	1	W	NACK_RECV_CLR	NACK_RECV_FLAG clearing bit 0：NA 1：Clearing NACK_RECV_FLAG
	0	W	SEQ_MATCH_CLR	SEQ_MATCH_FLAG Clearing bit 0：NA 1：Clearing SEQ_MATCH_FLAG
CTL_REG_32 (0x20)	3	R	ACK_RECV_FAILED_FLAG	In TX resend mode, when the response packet is successfully received, the value of NACK_RECV_FLAG or SEQ_MATCH_FLAG isn't set as 1, and with ACK_RECV_FAILED_EN enabling, the value of ACK_RECV_FAILED_FLAG is set to 1 0：The tx resend mode successfully receive the response packet. 1：The tx resend mode failed to receive the response packet.
	2	R	TX_RESEND_DONE_FLAG	The TX ACK mode provides the resend function. Users can configure the resend numbers. When the number of resend reaches the specified number but no response or error received, while the TX_RESEND_DONE_EN function is enabled, the TX_RESEND_DONE_FLAG is set to 1 0：tx resend unfinished 1：tx resend finished
	1	R	NACK_RECV_FLAG	In TX resend mode, when a reply packet is received and NACK_RECV_EN is enabled, the received NACK bits are mapped to NACK_RECV_FLAG 0：The nack bit of the reply packet is 0 1：The nack bit of the reply packet is 1
	0	R	SEQ_MATCH_FLAG	In TX resend mode, after SEQ_MATCH_EN is enabled, the system compares the received seq_num with the local seq_num. If the matching and SEQ_MATCH_EN is enabled, the SEQ_MATCH_FLAG is set to 1

CMOSTEK Confidential

Register Name	Bit	R/W	Bit Flag	Function Description
				0：The local seq_num does not match the received seq_num 1：The local seq_num matches the received seq_num
CTL_REG_84 (0x54)	6	RW	FCS2_EN	Configure whether the packet has the FCS2 field 0：packet without fcs2 field 1：packet with fcs2 field
	5	RW	SEQNUM_MATCH_EN	Seqnum_en and tx ACK mode enabled. This bit compares the local seq_num with the received seq_num 0：disable seq_num comparison 1：enable seq_num comparison
	4	RW	SEQNUM_SIZE	Configure the length of seq_num 0：1 byte 1：2 bytes
	3	RW	SEQNUM_AUTO_INC	Whether SEQNUM is automatically incremented in TX 0：Not accumulated 1：Automatically incremented by 1 each packet
	2	RW	SEQNUM_EN	Configure whether the packet contains the seqnum field 0：packet without seqnum field 1：packet with seqnum field
CTL_REG_87 (0x57)	7:0	RW	SEQNUM_TX_IN[7:0]	Seqnum lower 8 bit，users can configure seqnum in tx ack mode by themselves.
CTL_REG_88 (0x58)	7:0	RW	SEQNUM_TX_IN[15:8]	Seqnum higher 8 bit，users can configure seqnum in tx ack mode by themselves.
CTL_REG_38 (0x26)	7:0	R	SEQNUM_TX_OUT[7:0]	Lower 8 bits of the seqnum of the actual tx packet.
CTL_REG_39 (0x27)	7:0	R	SEQNUM_TX_OUT[15:8]	Higher 8 bits of the seqnum of the actual tx packet.
CTL_REG_91 (0x5B)	7:0	RW	FCS2_TX_IN[7:0]	Users can self-configure the value of the fcs2 field.
CTL_REG_92 (0x5C)	7:0	R	FCS2_RX_OUT[7:0]	Received the value of the fcs2 field of the packet

CMOSTEK Confidential

The resend function is an automatic control mechanism for the handshake between the transmitter and receiver. This function provides two Packet format fields for determining whether the handshake between the transmitter and receiver is successful, namely, SEQNUM and FCS2 fields, which are part of Packet. The function has to be enabled through

SEQNUM_EN and FCS2_EN.

The implementation principle is that the sender configures the local SEQNUM and transmits the NACK bit in the FCS2 field to 1. After the packet is successfully sent, the sender enters the RX state and waits for reply. When the received reply packet SEQNUM matches the local SEQNUM and the received NACK bit as 1, it is thought to shake hands successfully, which means the receiver have successfully received the transmitted packet.

After the ACK function is enabled on the receiving end, it enters the RX state. When the packet is successfully received, the NACK bits are used to determine whether the packet needs to be answered. If the packet needs to be answered, the packet enters the TX state and sends back the received serial number and NACK bit. PKT_DONE interruption occurs when the sending back complete. After that, PKT_DONE_EXIT_EN is used to determine whether to return to the RX_EXIT_STATE configuration state or remain in the RX state.

5.1.1 TX ACK Function Usage Description

The TX ACK mode provides 4 interrupt flags for users to query the current status of the chip, which includes ACK_RECV_FAILED_FLAG, TX_RESEND_MAX_FLAG, NACK_RECV_FLAG and SEQ_MATCH_FLAG. When ACK_RECV_FAILED_EN, TX_RESEND_MAX_EN, NACK_RECV_EN and SEQ_MATCH_EN is set to 1 respectively, the corresponding events will occur and the corresponding interrupt flag bit will be set to 1.

ACK_RECV_FAILED_FLAG is set to 1 when the chip received an ACK packet with unmatched compared information. While The TX_RESEND_DONE_FLAG value is set to 1 when the chip does not receive a reply packet or receives an incorrect reply packet after executing the TX_RS_TIMES number of re-transmission times. The NACK_RECV_FLAG value depends on the NACK bit in the received ACK packet in FCS2 field. SEQ_MATCH_FLAG is set to 1 when SEQ_MATCH_EN is enabled and the response packet SEQNUM is the same as the local SEQNUM

The ACK mode provides two Packet format fields for determining whether the handshake between the transmitter and receiver is successful, namely, SEQNUM and FCS2 fields, which are part of Packet. The function has to be enabled through SEQNUM_EN and FCS2_EN. The implementation principle is that the sender configures the local SEQNUM and transmits the NACK bit in the FCS2 field when it is set as 1. After the packet is successfully sent, the sender enters the RX state and waits for reply. When the received reply packet SEQNUM matches the local SEQNUM and the received NACK bit as 1, it is thought to shake hands successfully, which means the receiver have successfully received the transmitted packet.

Whether it needs to have SEQNUM matching is controlled through SEQNUM_MATCH_EN. When it is enabled, it will have matching process when the ACK packet is received. If it is matched, SEQ_MATCH_FLAG will be set to 1; if it is disabled, there's no matching process. The SEQNUM size can be configured through SEQNUM_SIZE with a range of 1 - 2 bytes. The SEQNUM value can be initialized through the configuration of SEQNUM_TX_IN<15:0>. SEQNUM is capable of increasing by 1 automatically, which can be enabled through SEQNUM_AUTO_INC. When it is enabled, every time before it transmits, the SEQNUM will increase by 1 based on the original value, after then it transmits out the data. Users can query the current transmitted SEQNUM value through SEQNUM_TX_OUT<15:0>.

NACK bit is the highest bit of FCS2 with other bits reserved. FCS2 field can be configured through FCS2_TX_IN<7:0>. If FCS2

CMOSTEK Confidential

field is enabled, the received FCS2 value will be put into register FCS2_RX_OUT<7:0> as packet received,

In TX ACK mode, RX T1 TIMER is involved in the automatic processing control. Therefore TIMER_RX_EN must be set to 1 when this mode is used. RX T1 TIMER is configured through TIMER_M_RX_T1 and TIMER_R_RX_T1 and the actual counting time is calculated through the formula. If PKT_DONE is received successfully, the TX_EXIT_STATE is returned. If not, the TX_EXIT_STATE is returned to READY and ACK_RECV_FAILED_FLAG is set to 1

If no reply packet is successfully received, users can re-enter the TX and wait for the reply packet in TX RESEND mo de. The number of resending times is configured through TX_RESEND_TIMES<7:0>. When no reply packet is received after the maximum number of retransmissions is reached, the TX_RESEND_DONE_FLAG position 1 and TX_RESEND_ DONE_TIMES<7:0> indicates the numbers of resending have been completed.

After finishing configurations, users can set TX_ACK_EN to 1 to enable it. After enabling, users can send go_tx command to enter the TX ACK mode. During automatic operating, if triggers stop events, the chip will exit automatic operating and stops in a fixed state. If users would like to stop the TX ACK function, they can set API_STOP to 1. When the chip detects this bit, it will set API_DONE_FLAG to 1 upon next state switching and keep current configuration and then exits TX ACK to return to READY state.

CMOSTEK Confidential

5.1.2 TX RESEND Sequence Chart Description

1. ACK packet is not received successfully when the number of maximum resending times is reached.

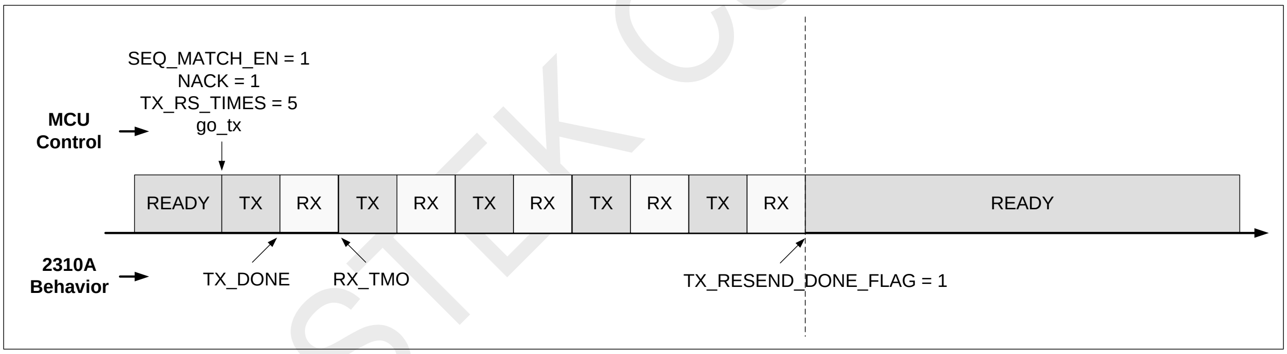


Figure 39. No Packet Reply when TX RESEND Reaches the Maximum Resending Times

2. ACK packet received successfully with unmatched content.

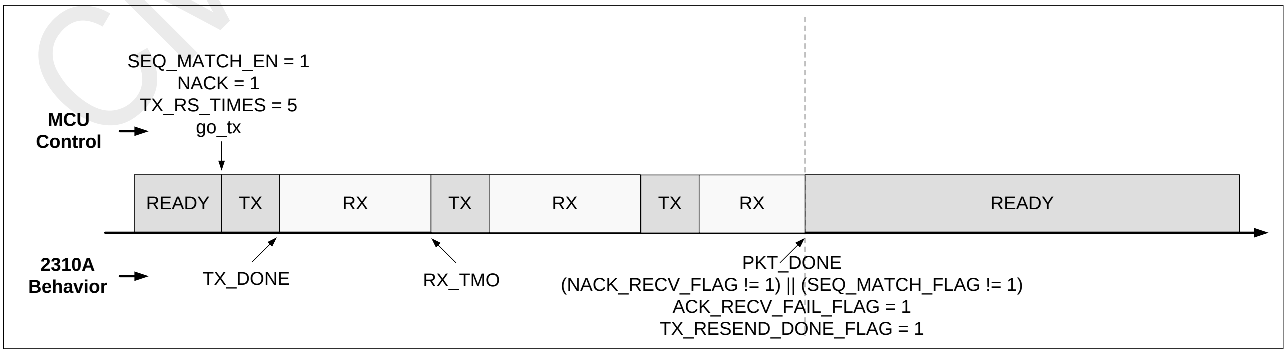


Figure 40. ACK Packet is Received but Unmatched in TX RESEND

3. ACK packet is received successfully and the content is matched.

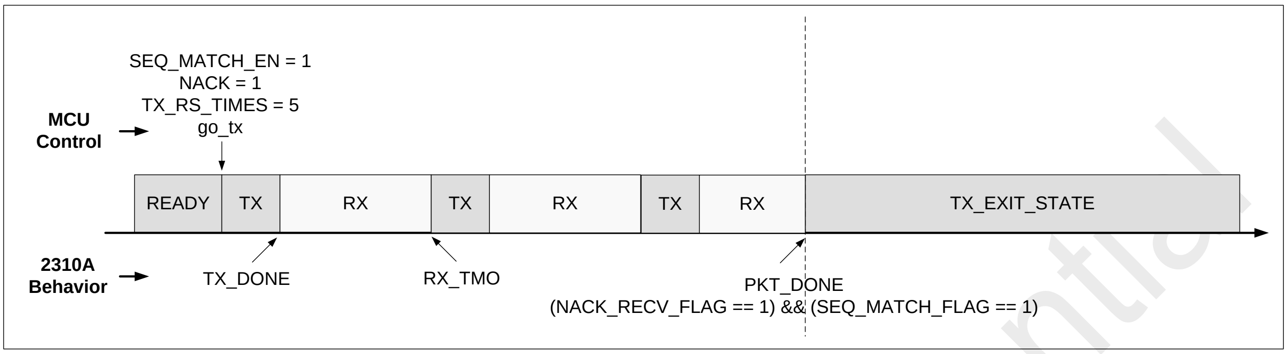


Figure 41. ACK Packet is Received with Matched Content in TX RESEND

4. TX_RESEND is combined with TX_DUTY_CYCLE. An error ACK packet is received during the waiting process and exit with failure.

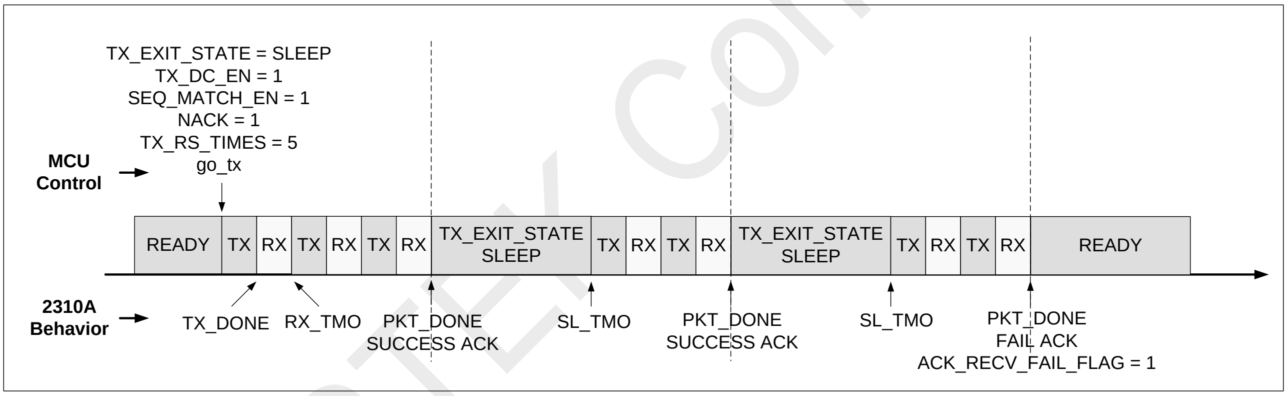
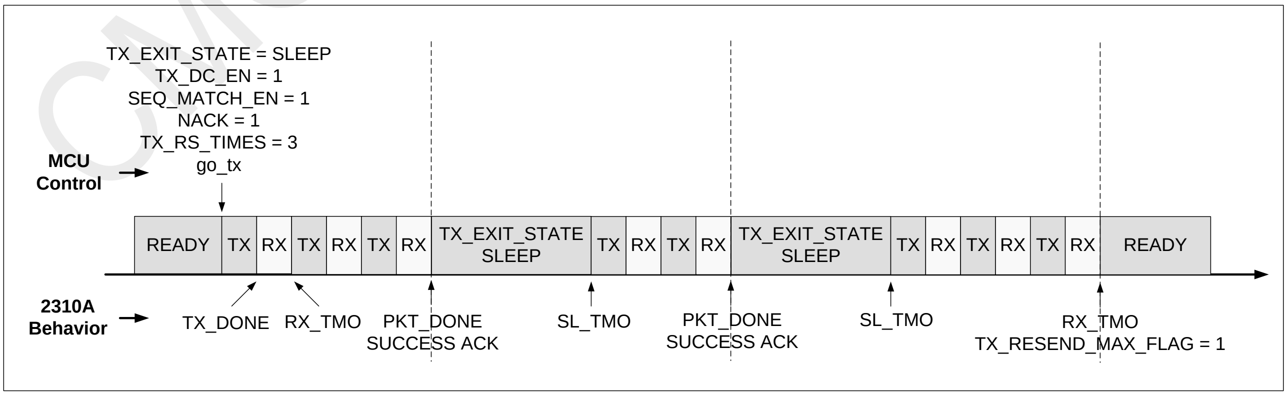


Figure 42. Error ACK Packet Received during the Waiting Process and Exit with Failure

5. TX_RESEND is combined with TX_DUTY_CYCLE. When it reaches the maximum numbers, it exits with failure.



CMOSTEK Confidential

Figure 43. Exit with Failure when the Resending Times reach the Maximum Numbers

6. TX_RESEND is combined with TX_DUTY_CYCLE, and the ACK packet is received successfully during the process, and it exits after reaching TX_DC_TIMES

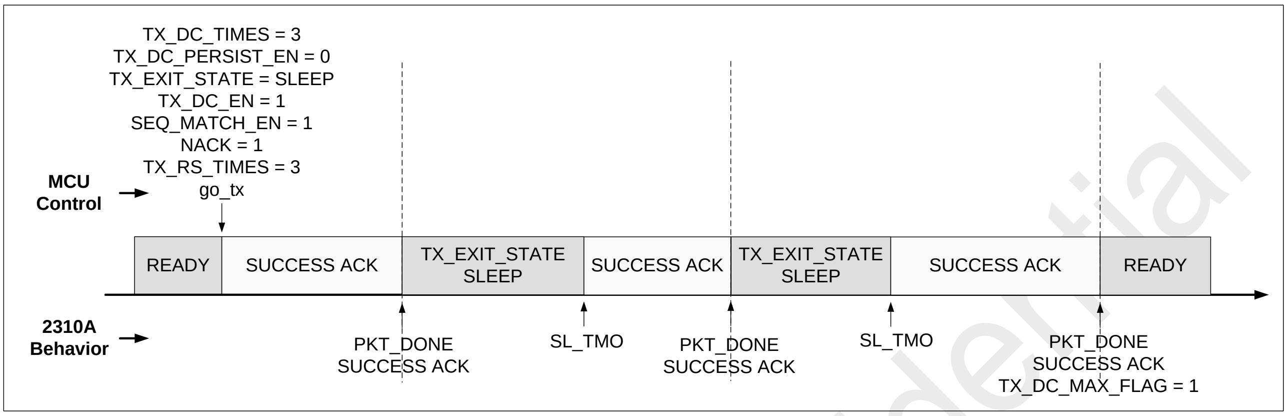


Figure 44. Exit after Reaching the TX_DC_TIMES

7. TX_RESEND binds to TX_DUTY_CYCLE and exits via api_stop in TX DC PERSISTENT mode.

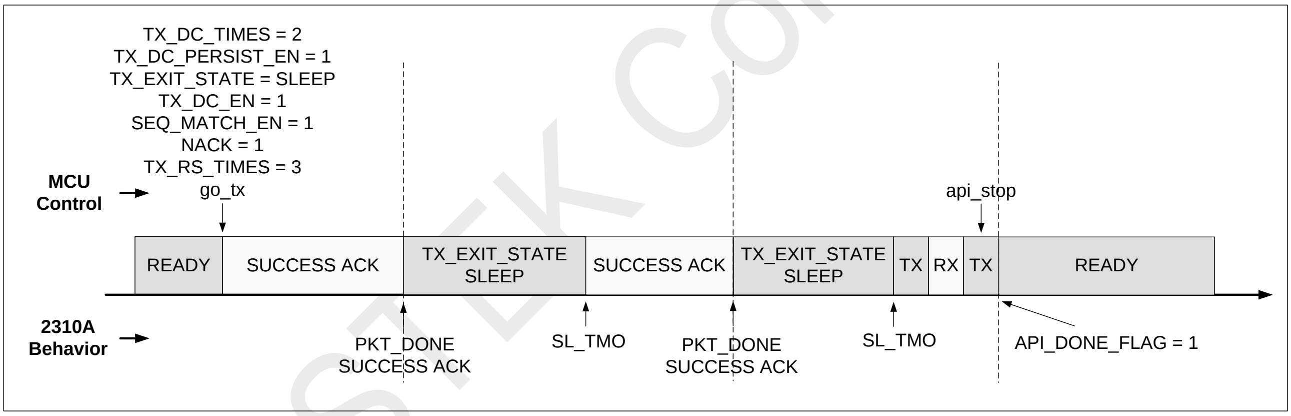
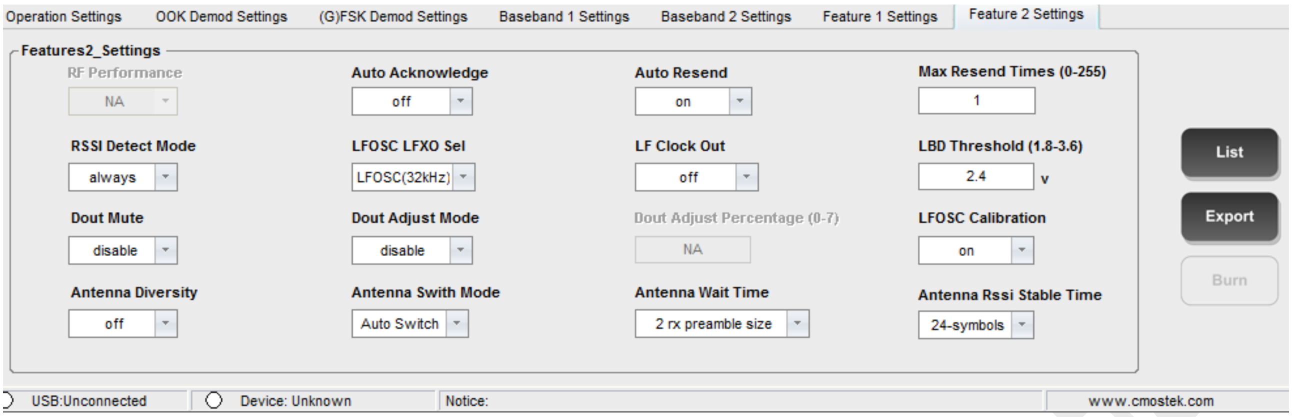


Figure 45. Exit through api_stop under TX DC PERSISTENT Mode

5.2 RX ACK

The corresponding RFPDK interface and parameters are shown as followed.

CMOSTEK Confidential



Features2_Settings

RF Performance: NA
Auto Acknowledge: off
Auto Resend: on
Max Resend Times (0-255): 1
RSSI Detect Mode: always
LFOSC LFXO Sel: LFOSC(32kHz)
LF Clock Out: off
LBD Threshold (1.8-3.6): 2.4 v
Dout Mute: disable
Dout Adjust Mode: disable
Dout Adjust Percentage (0-7): NA
LFOSC Calibration: on
Antenna Diversity: off
Antenna Swith Mode: Auto Switch
Antenna Wait Time: 2 rx preamble size
Antenna Rssi Stable Time: 24-symbols

USB:Unconnected
Device: Unknown
Notice:
www.cmostek.com

Figure 46. The Related Parameters on RFPDK Interface under RX ACK

CMOSTEK Confidential

Table 25. TX Auto Resend Related Parameters

Register-bit RFPDK Parameter	Register Bit
Auto Acknowledge	RX_ACK_EN

Related registers and descriptions are shown under the table below.

Table 26. Related Registers in RX ACK

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_84 (0x54)	6	RW	FCS2_EN	Configure whether the packet has fcs2 field. 1：packet with fcs2 field 0：packet without fcs2 field
	4	RW	SEQNUM_SIZE	Configure length of seq_num 1：2 bytes 0：1 byte
	2	RW	SEQNUM_EN	Configure whether the packet has seqnum field. 1：packet with seqnum field 0：packet without seqnum field
CTL_REG_87 (0x57)	7:0	RW	SEQNUM_TX_IN[7:0]	seqnum lower 8 bit. Users can configure seqnum in tx ack mode.
CTL_REG_88 (0x58)	7:0	RW	SEQNUM_TX_IN[15:8]	Seqnum higher 8 bit，Users can configure seqnum in tx ack mode.
CTL_REG_38 (0x26)	7:0	R	SEQNUM_TX_OUT[7:0]	The actual lower 8 bit of seqnum in tx packet.
CTL_REG_39 (0x27)	7:0	R	SEQNUM_TX_OUT[15:8]	The actual higher 8 bit of seqnum in tx packet.
CTL_REG_91	7:0	RW	FCS2_TX_IN[7:0]	Users can configure the value of the fcs2

Register Name	Bit	R/W	Bit Flag	Function Description
(0x5B)				field.
CTL_REG_92 (0x5C)	7:0	RW	FCS2_RX_OUT[7:0]	Receive the fcs2 field value of the packet.
CTL_REG_97 (0x61)	6:4	RW	RX_EXIT_STATE[3:0]	The state after Rx timer timeout.
	3	RW	TIMER_RX_EN	rx timer enabling bit, rx timer starts working after enabling as the configuration. 1：enable 0：disable
	2	RW	RX_ACK_EN	rx ack functional enabling bit. 1：enable 0：disable
CTL_REG_98 (0x62)	7	RW	PKT_DONE_EXIT_EN	Whether to remain in the current state or exit immediately to the state defined under RX_EXIT_STATE when receive the packet successfully and generate pkt_done. 0：the chip is remained in the current state 1：the chip will back to the state defined under RX_EXIT_STATE
CTL_REG_101 (0x65)	7:0	RW	RX_TIMER_T1_M<7:0>	The counting time of RX T1 TIMER is cal-culated through the formula of： T = M x 2^(R+1) x 20 us The value range of R is 0-21.
CTL_REG_102 (0x66)	7:5	RW	RX_TIMER_T1_M<10:8>	
	4:0	RW	RX_TIMER_T1_R<4:0>	

5.2.1 Description of Functional Usage in RX ACK

After the receiver is enabled and receives the packet, it first checks whether the NACK bit in the FCS2 field is 1. If it is 1, it switches back to TX and sends the received SEQNUM and the self-configured FCS2 field back to the sender.It can be a simple packet with SEQNUM and FCS2 field without payload, or a riding packet with SEQNUM and FCS2 field with payload. It should be noted that in the case of riding packet, both the sender and receiver should be configured in variable length mode. The PIGGYBACKING configuration decides whether it is a simple packet or a riding packet. If it is configured as 1 then it is a riding packet with content from TX FIFO. Therefore, TX FIFO content has to be filled before transmission.

The size of SEQNUM can be set through SEQNUM_SIZE with range from 1 to 2 bytes. SEQNUM can be initialized by users through SEQNUM_TX_IN<15:0>. SEQNUM_TX_OUT<15:0> is used for query the current transmission SEQNUM value. The SEQNUM received will be stored into RX_SEQNUM<15:0> for user reading.

The NACK bit is the highest bit in the FCS2 field as the other bits are reserved bit. The FCS2 field can be configured through fcs2_tx_in<7:0>. If the FCS2 field is enabled, the received packet will put the FCS2 field content into

CMOSTEK Confidential

FCS2_RX_OUT<7:0>.

The automatic process of RX ACK mode involves the RX T1 TIMER, so TIMER_RX_EN must be set to 1 while operation. RX T1 TIMER can be configured through TIMER_M_RX_T1 and TIMER_R_RX_T1 and the actual counting time can be calculated from the formula.

In the case of post back, the receiver PKT_DONE is generated only after the post back complete. PKT_DONE_EXIT_EN determines whether to exit and return to the RX_EXIT_STATE configured state or remain in the RX state.

After users complete the configuration, RX_ACK_EN is enabled by 1. After enabling, go_rx command is sent to enter the RX ACK mode. If the stop event is triggered during the automatic operation, the chip will exit the automatic process and stop to a fixed state. If users would like to stop the RX ACK function, API_STOP can be set to 1. After detecting this bit, the chip will set API_DONE_FLAG to 1 upon the next state switch and keep the current configuration and back to READY state.

CMOSTEK Confidential

5.2.2 RX ACK Sequence Diagram Description

1. Receive packet in RX_ACK with acknowledge bit as 0 and PKT_DONE_EXIT_EN = 1

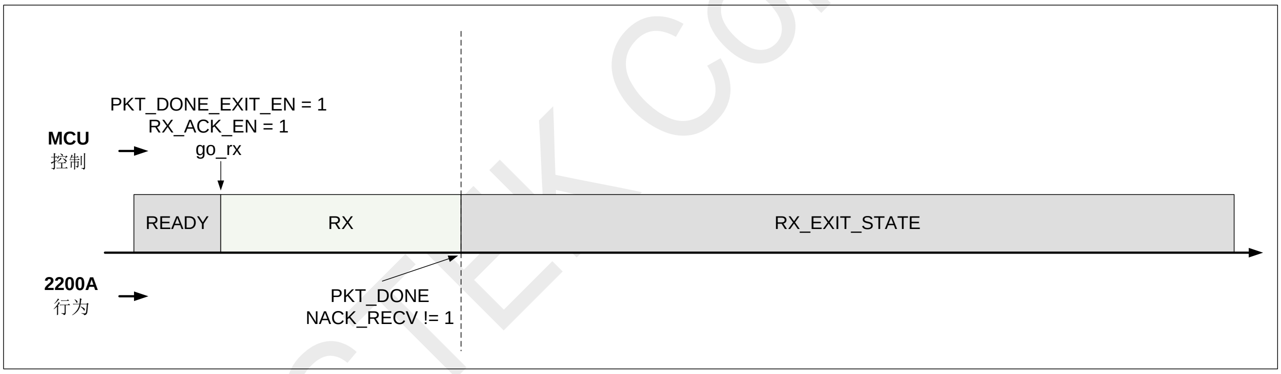


Figure 47. NACK = 0，PKT_DONE_EXIT_EN = 1

2. Receive packet in RX_ACK with acknowledge bit as 0 and PKT_DONE_EXIT_EN = 0

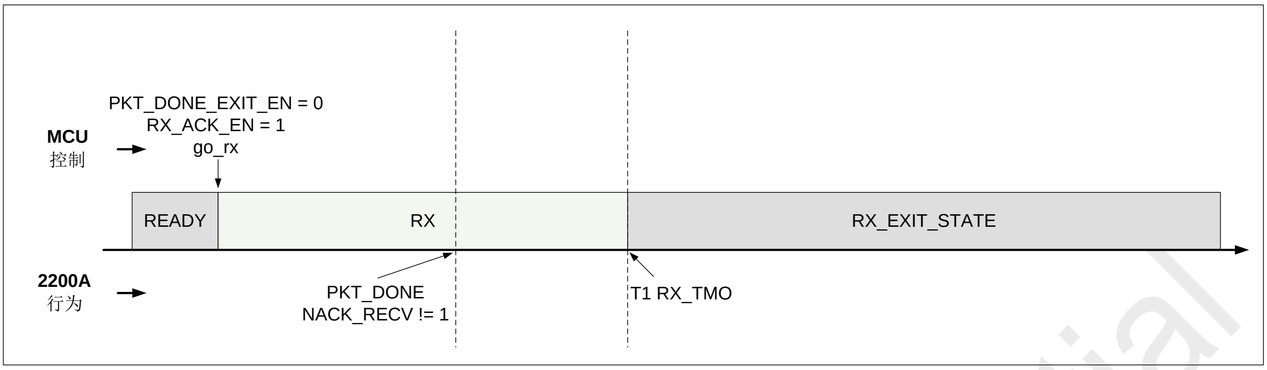


Figure 48. NACK = 0，PKT_DONE_EXIT_EN = 0

3. Did not received acknowledge packet in RX_ACK，exit when it is RX TIMEOUT.

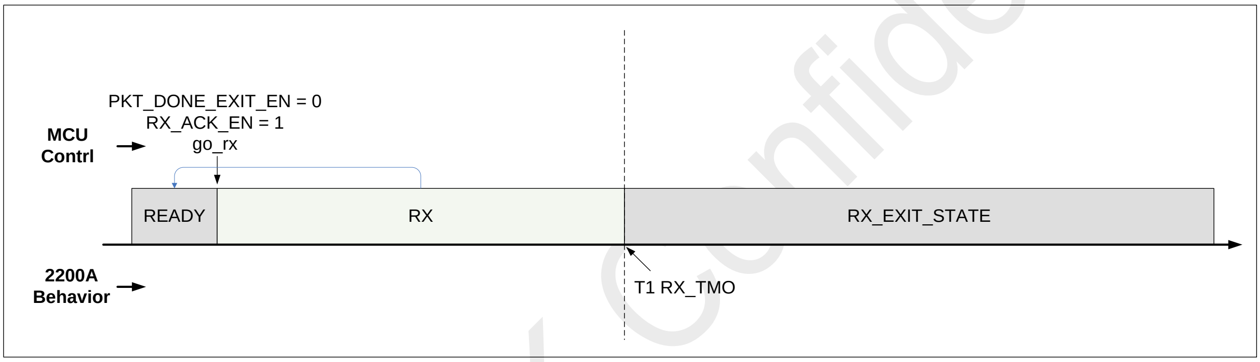


Figure 49. Packet not Received in RX_ACK，Exit RX TIMEOUT

4. Receive packet in RX_ACK with acknowledge bit as 1 and PKT_DONE_EXIT_EN = 0

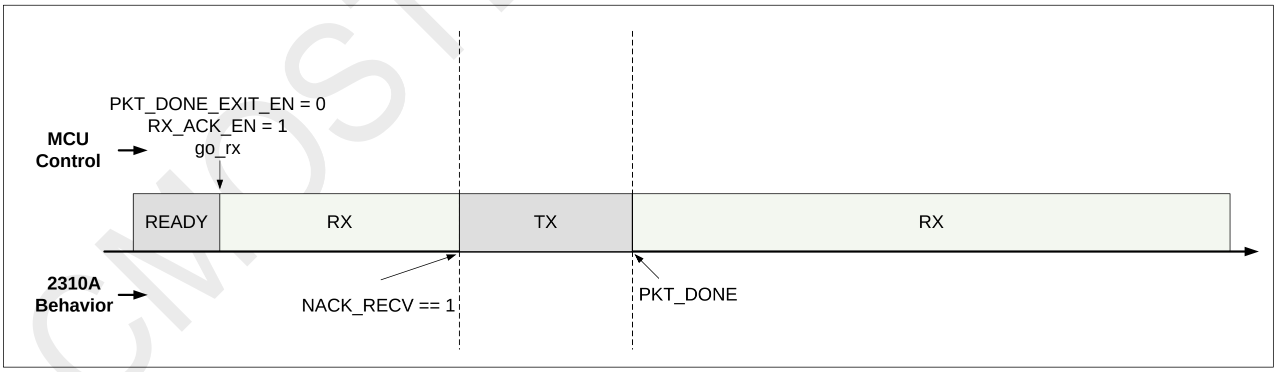


Figure 50. NACK = 1，PKT_DONE_EXIT_EN = 0

5. Receive packet in RX_ACK with acknowledge bit as 1 and PKT_DONE_EXIT_EN = 1

CMOSTEK Confidential

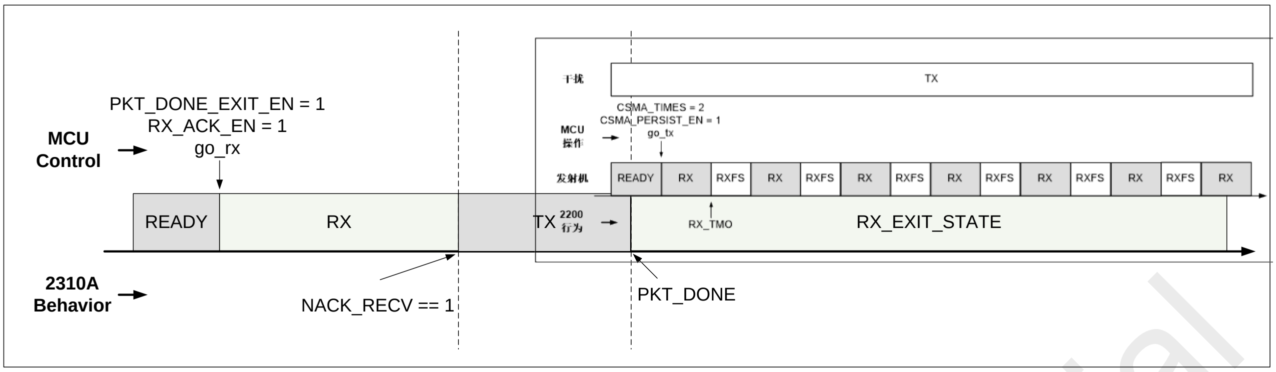
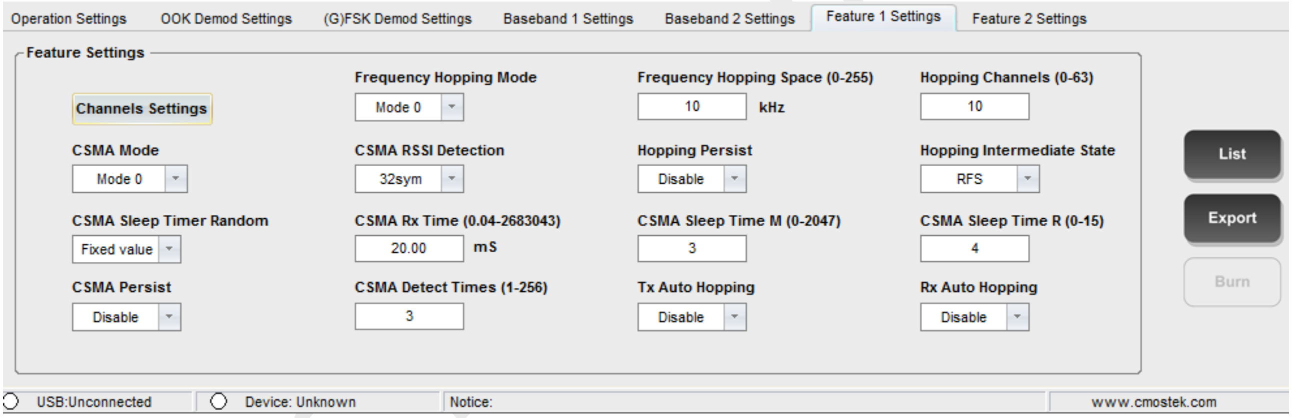


Figure 51. NACK = 1，PKT_DONE_EXIT_EN = 1

6. Carrier Sense Multiple Access（CSMA）

6.1 Related registers in CSMA Mode

The corresponding RFPDK interface and parameters are shown as followed.



Frequency Hopping Mode: Mode 0

Frequency Hopping Space (0-255): 10 kHz

Hopping Channels (0-63): 10

CSMA Mode: Mode 0

CSMA RSSI Detection: 32sym

Hopping Persist: Disable

Hopping Intermediate State: RFS

CSMA Sleep Timer Random: Fixed value

CSMA Rx Time (0.04-2683043): 20.00 mS

CSMA Sleep Time M (0-2047): 3

CSMA Sleep Time R (0-15): 4

CSMA Persist: Disable

CSMA Detect Times (1-256): 3

Tx Auto Hopping: Disable

Rx Auto Hopping: Disable

Figure 52. CSMA Related Parameters on RFPDK

Table 27. CSMA Related Parameters

Register-bit RFPDK Parameter	Register Bit
CSMA Mode	CCA_MODE<2:0>
CSMA RSSI Detection	CCA_WIN_SEL<1:0>
CSMA Sleep Timer Random	TIMER_RANDOM_MODE<1:0>
CSMA Rx Time	TIMER_M_RX_CSMA<10:0> TIMER_R_RX_CSMA<4:0>
CSMA Sleep Time M	TIMER_M_SLEEP_CSMA<10:0>
CSMA Sleep Time R	TIMER_R_SLEEP_CSMA<4:0>
CSMA Persist	CSMA_PERSIST_EN

CMOS nfidential

CSMA Detect Times	CSMA_TIEMS<7:0>

The related register description is as follows.

Table 28. Related Registers in Configuration Area

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_7 (0x07)	4	RW	API_STOP	The bit is configured to stop the feature function. After the feature is set to 1, it does not take effect immediately. Instead, the chip takes effect upon next state switch. Feature returns to the Ready state after it stops successfully 0: not stop feature. 1: stop feature.
CTL_REG_23 (0x17)	7	RW	API_DONE_EN	The enabling bit of both API_DONE_FLAG and API_DONE. After enabling, if the corresponding API_DONE_FLAG events are valid, the API_DONE_FLAG willbe set to 1 and generated API_DONE interruption accordingly. 0: disable. 1: enable.
	6	RW	CCA_STATUS_EN	The enabling bit of both CCA_STATUS_FLAG and CCA_STATUS. After enabling, if the corresponding CCA_STATUS_FLAG events are valid, the CCA_STATUS_FLAG will be set to 1 and generated API_DONE interruption accordingly. 0: disable. 1: enable.
	5	RW	CSMA_DONE_EN	The enabling bit of both CSMA_DONE_FLAG and CSMA_DONE. After enabling, if the corresponding CSMA_DONE_FLAG events are valid, the CSMA_DONE_FLAG will be set to 1 and generated CSMA_DONE interruption accordingly. 0: disable. 1: enable.
CTL_REG_31 (0x1F)	7	W	API_DONE_CLR	API_DONE_FLAG clearing bit 0：NA 1：Clearing API_DONE_FLAG
	6	W	CCA_STATUS_CLR	CCA_STATUS_FLAG clearing bit

CMOSTEK Confidential

Register Name	Bit	R/W	Bit Flag	Function Description
				0：NA 1：Clearing CCA_STATUS_FLAG
	5	W	CSMA_DONE_CLR	CSMA_DONE_FLAG clearing bit 0：NA 1：Clearing CSMA_DONE_FLAG
CTL_REG_32 (0x20)	7	R	API_DONE_FLAG	The API_DONE_FLAG bit. When the api_stop bit is set to 1 and respond successfully, the chip will return to the READY state. The chip will set API_DONE_FLAG to 1 when API_DONE_EN is enabled. 0: api_stop is not responded successfully. 1: api_stop is responded successfully and feature is stopped.
	6	R	CCA_STATUS_FLAG	The CCA_STATUS_FLAG bit. When csma function isapplied, it enters RX state first to detect whether the current channel is in idle state. If current channel is busy and CCA_STATUS_EN is enabled, then CCA_STATUS_FLAG will be set to 1. 0: current channel is idle. 1: current channel is busy.
	5	R	CSMA_DONE_FLAG	The maximum number of signal channel sensing timesthat can be configured in csma function. If the signal channel is not in idle yet when the maximum number is reached as CSMA_DONE_EN is enabled, then CSMA_DONE_FLAG will be set as 1. 0: the maximum number of signal channel sensing times is not reached. 1: the maximum number of signal channel sensing times is reached.
CTL_REG_96 (0x60)	6:4	RW	TX_EXIT_STATE[2:0]	The returning state after TX is finished. 1：SLEEP 2：READY 3：TFS 4：TX 5：RFS 6：RX Others：SLEEP
CTL_REG_97 (0x61)	7	RW	CSMA_EN	Csma functional enabling bit 0: disable. 1: enable.
	3	RW	TIMER_RX_EN	rx timer enabling bit 0: disable.

CMOSTEK Confidential

Register Name	Bit	R/W	Bit Flag	Function Description
				1: enable.
CTL_REG_105 (0x69)	5:4	RW	TIMER_RANDOM_MODE[1:0]	The csma sleep timer mode under csma mode. 00: R value is random. 01: M value is random. 10: both R and M are random. 11: use the fixed value configured by users.
	3	RW	TIMER_SLEEP_EN	sleep timer enabling bit 0: disable. 1: enable.
CTL_REG_105 (0x6A)	7:5	RW	CCA_MODE[2:0]	Configuration of cca_status 000：recognize the signal channel to be idle all the time. 001：If RSSI_OK is detected at least once among the four detection window, the channel is considered busy 010：If PJD_OK is detected at least once among the four detection window, the channel is considered busy 011：If RSSI_OK or PJD_OK is detected at least once among the four detection window, the channel is considered busy. 100：If one SYNC_PASS is detected then the channel is considered busy 101：If SYNC_PASS is detected once, or RSSI_OK is detected at least once in four detection Windows, the channel is considered busy. 110：If SYNC_PASS is detected once, or PJD_OK is detected at least once in four detection Windows, the channel is considered busy. 111：If SYNC_PASS is detected once, or RSSI_OK or PJD_OK is detected at least once in four detection Windows, the channel is considered busy.
	4	RW	CSMA_PERSIST_EN	After the bit is set to 1, the chip detects the idle channel until the packet is successfully sent, regardless of the CSMA_times configuration. 1：Continuously detect until the signal channel is idle and transmit successfully 0：csma didn’t keep detecting
	1:0	RW	CCA_WIN_SEL[1:0]	Size of detecting window of csma 00：32 symbols 01：64 symbols 10：128 symbols 11：256 symbols
CTL_REG_107 (0x6B)	7:0	RW	TIMER_M_RX_CSMA [7:0]	The counting time of rx csma timer is calculated as the formula of：

CMOSTEK Confidential

Register Name	Bit	R/W	Bit Flag	Function Description
CTL_REG_108 (0x6C)	7:5	RW	TIMER_M_RX_CSMA [10:8]	T = M x 2^(R+1) x 5us
	4:0	RW	TIMER_R_RX_CSMA [4:0]	
CTL_REG_115 (0x73)	7:0	RW	CSMA_TIMES [7:0]	Configure the maximum detection number of the csma channel, with a range of 0-255.
CTL_REG_116 (0x74)	7:0	RW	CSMA_DONE_TIMES [7:0]	The already detection numbers of csma，with a range of 0-255.
CTL_REG_118 (0x76)	7:0	RW	TIMER_M_SLEEP_CSMA [7:0]	The counting time of sleep csma timer is calculated as the formula of： T = M x 2^(R+1) x 31.25us
CTL_REG_119 (0x77)	7:5	RW	TIMER_M_SLEEP_CSMA [10:8]	
	4:0	RW	TIMER_R_SLEEP_CSMA [4:0]	

6.2 Functional Usage Description

CSMA is a conflict preventing mechanism that detects whether the current signal is in busy or idle state before transmission and then decides whether to conduct transmission, that is listening before talking, which avoid the situation of different transmitters use the same signal channel and to increase the probability of successful receiving data packet at receiver side. The core of this mechanism is to judge the signal channel situation through comparison between RSSI and the preset threshold. When the signal channel is busy, the process of sensing back-off will repeat for certain times until the signal channel is idle. If it still fails after the limited times reached, then the corresponding interrupts and flags will be generated and transmission will be stopped. During the procedure of back-off, the chip stays in SLEEP state. However, in the PERSISTENT mode of CSMA, it does not back off to SLEEP state, but instead returns to RFS state and switches to RX state for sensing. In this mode, it will continue detecting the signal channel until the data is transmitted out. In NON-PERSISTENT mode, the number of maximum sensing times at non- idle channel can be configured through CSMA_TIMES. If the signal channel is still busy when the maximum sensing times is reached and CSMA_DONE_FLAG is configured to 1, then it will exit CSMA and back to READY state.

The CSMA mode provides two interrupt flags for users to query current chip status, which includes CCA_STATUS_FLAG and CSMA_MAX_FLAG. When CCA_STATUS_EN and CSMA_MAX_EN are set as 1, CCA_STATUS_FLAG / CSMA_MAX_FLAG will be set as 1 as the corresponding events occur. CCA_STATUS_FLAG will be set as 1 when the chip sensed channel busy after enters RX. CSMA_DONE_FLAG will be set as 1 when the channel is still busy after performing the CSMA_TIMES configuration or after the channel signal has been successfully sent.

In CSMA mode, both RX TIMER and SLEEP TIMER are involved. When using CSMA function, RX TIMER and SLEEP TIMER must be set as 1. In addition, CSMA mode provides independent configuration of RX TIMER and SLEEP TIMER. RX TIMER is configured through TIMER_M_RX_CSMA and TIMER_M_RX_CSMA with the actual counting time being calculated by the formula; SLEEP TIMER. is configured through TIMER_M_SLEEP_CSMA and TIMER_M_SLEEP_CSMA with the actual counting time being calculated by the formula.

CMOSTEK Confidential

CSMA is categorized with PERSISTENT mode and NON-PERSISTENT mode through the configuration of CSMA_PERSIST_EN. In NON-PERSISTENT mode, CSMA_TIMES is the maximum sensing times, CSMA_DONE_TIMES is the times of finished sensing which increases by 1 each time. If the signal channel is still busy when the number of maximum times of CSMA_TIMES is reached, CSMA_MAX_FLAG will be set as 1 and exits CSMA. In PERSISTENT mode, regardless of the relation of CSMA_TIMES and CSMA_DONE_TIMES, the chip will keep sensing until the signal channel becomes idle and transmits the data out.

Whether the signal channel is busy or not is determined by CCA_MODE. See the description of register CTL_REG_105 (0x6A) for more detail information. In CSMA mode, there are multiple signal channel sensing window with each window size depending on CCA_WIN_SEL. See the description of register CTL_REG_105 (0x6A) for more details.

In CSMA mode, the counting time of SLEEP TIMER depends on both TIMER_M_SLEEP_CSMA and TIMER_M_SLEEP_CSMA, which is not fixed. With TIMER_RANDOM_MODE, it can configure a random time to SLEEP TIMER. See the description of register 0CTL_REG_105 (0x69) for more details.

After having the configurations down, users can set CSMA_EN as 1 to enable it, then send go_tx command to enter CSMA mode. During automatic operating, if the events that trigger stop operation, the chip will exit automatic operating and stops in a fixed state. If users would like to stop the CSMA function immediately, they can set API_STOP to 1. When the chip detects this bit, it will set API_DONE_FLAG to 1 upon state switching and keep the current configuration. Meanwhile it exits CSMA and returns to READY state.

If a packet is successfully received in the RX state of the CSMA, the chip does not generate PKT_DONE, so the chip behavior is not affected by PKT_DONE_EXIT_EN. Therefore, in CSMA mode, the chip must go through the complete RX window, and then judge the current channel condition after completing the entire RX window.

6.3 Description of Flowchart and Sequence Diagram

6.3.1 CSMA Flowchart

CMOSTEK Confidential

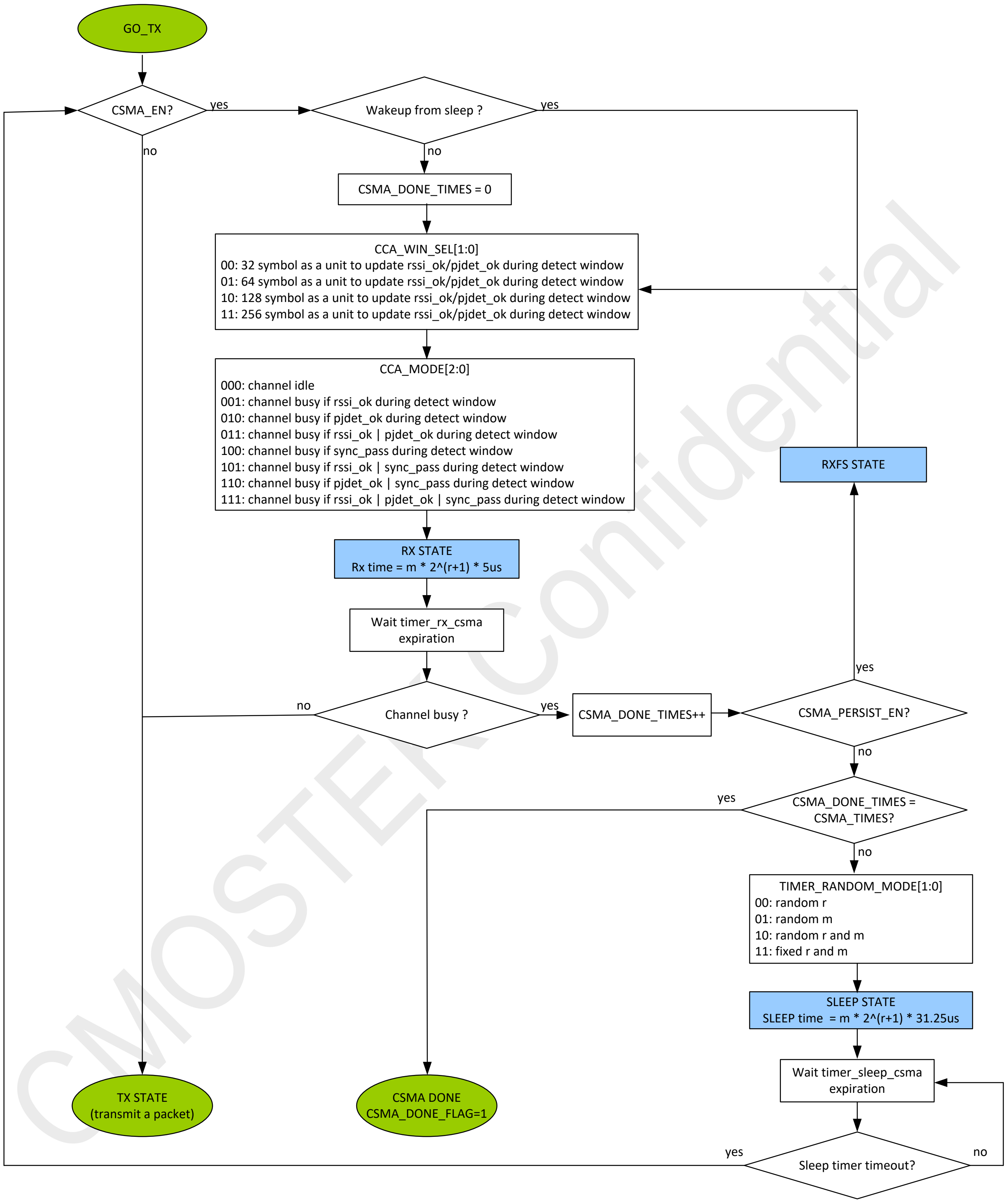

GO_TX
CSMA_EN?
yes
no
Wakeup from sleep ?
yes
no
CSMA_DONE_TIMES = 0
CCA_WIN_SEL[1:0]
00: 32 symbol as a unit to update rssi_ok/pjdet_ok during detect window
01: 64 symbol as a unit to update rssi_ok/pjdet_ok during detect window
10: 128 symbol as a unit to update rssi_ok/pjdet_ok during detect window
11: 256 symbol as a unit to update rssi_ok/pjdet_ok during detect window
CCA_MODE[2:0]
000: channel idle
001: channel busy if rssi_ok during detect window
010: channel busy if pjdet_ok during detect window
011: channel busy if rssi_ok | pjdet_ok during detect window
100: channel busy if sync_pass during detect window
101: channel busy if rssi_ok | sync_pass during detect window
110: channel busy if pjdet_ok | sync_pass during detect window
111: channel busy if rssi_ok | pjdet_ok | sync_pass during detect window
RXFS STATE
RX STATE
Rx time = m * 2^(r+1) * 5us
Wait timer_rx_csma expiration
yes
no
Channel busy ?
yes
CSMA_DONE_TIMES++
CSMA_PERSIST_EN?
no
yes
CSMA_DONE_TIMES = CSMA_TIMES?
no
TIMER_RANDOM_MODE[1:0]
00: random r
01: random m
10: random r and m
11: fixed r and m
SLEEP STATE
SLEEP time = m * 2^(r+1) * 31.25us
Wait timer_sleep_csma expiration
TX STATE
(transmit a packet)
CSMA DONE
CSMA_DONE_FLAG=1
yes
Sleep timer timeout?
no
CMOSTEK Confidential

Figure 53. CSMA Flowchart

6.3.2 CSMA Sensing Method

There are two types of sensing mode in CSMA: when CSMA_PERSIST_EN is 0, the chip will return to READY when it reaches its maximum sensing time as the channel is still busy. When CSMA_PERSIST_EN is 1, the chip will keep detecting the current channel until it is idle to send the data out.

1. CSMA_PERSIST_EN = 0，random sleep time，fail and exit when the maximum sensing times reached.

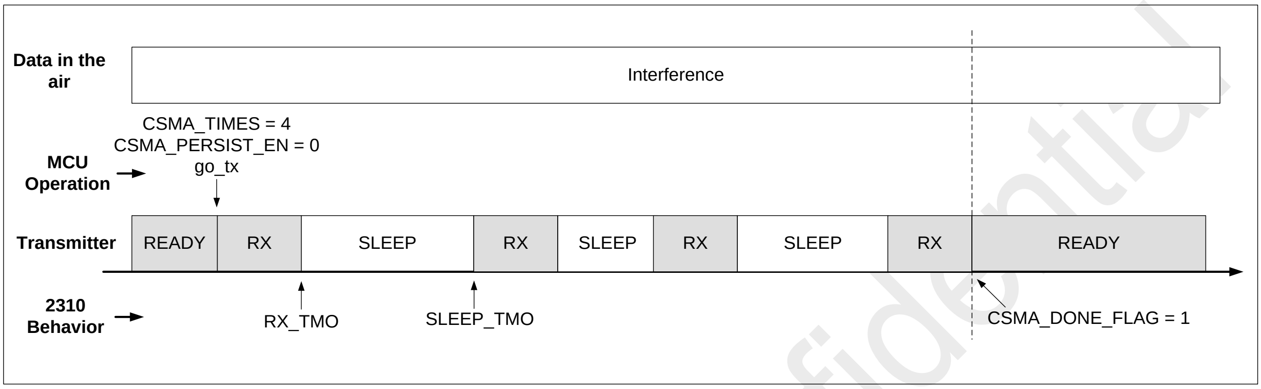


Figure 54. CSMA_PERSIST_EN = 0，Fail and Exit when the Maximum Sensing Times Reached

2. CSMA_PERSIST_EN = 0，random sleep time，TX successfully when the maximum detecting numbers isn’t reached.

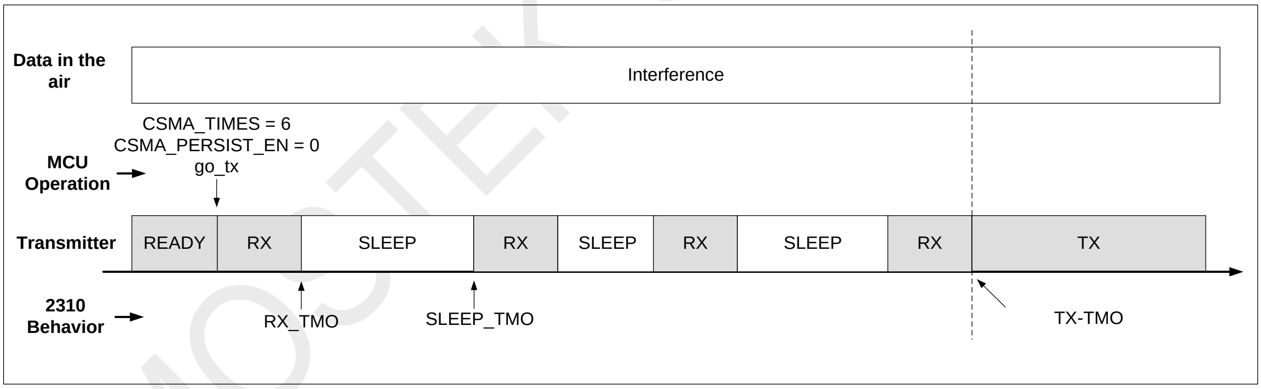


Figure 55. CSMA_PERSIST_EN = 0, TX Successfully when the Maximum Detecting Number isn’t Reached.

CMOSTEK Confidential

3. CSMA_PERSIST_EN = 1，signal channel keep busy all the time.

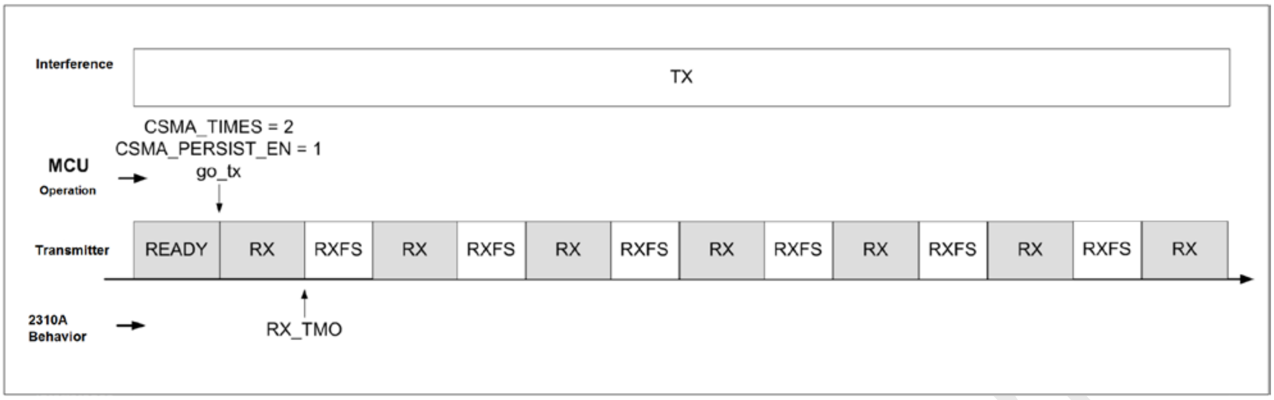


Figure 56. CSMA_PERSIST_EN = 1, Signal Channel Keep Busy

4. CSMA_PERSIST_EN = 1，successfully send out data until channel idle.

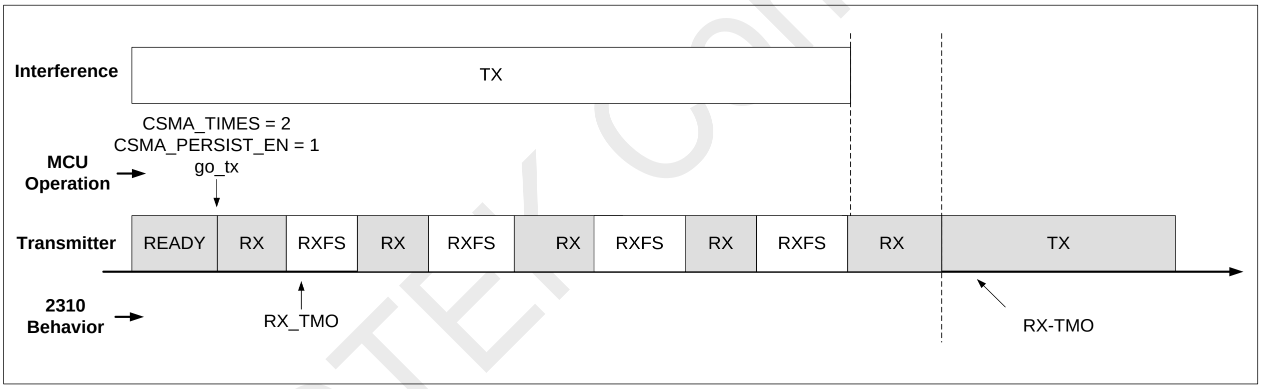


Figure 57. CSMA_PERSIST_EN = 1，Successfully Send out Data until Channel Idle

5. CSMA_PERSIST_EN = 1，signal channel keep busy and users exit via API_STOP

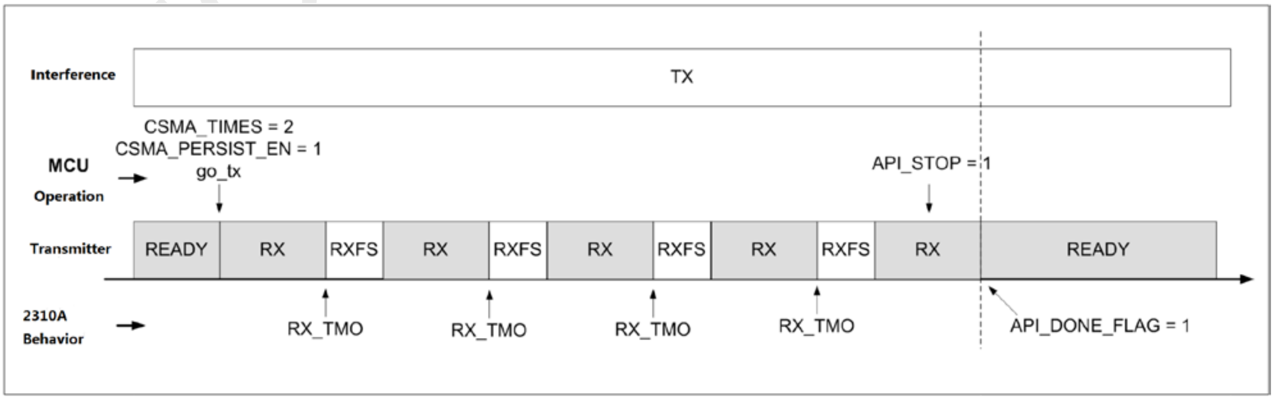


Figure 58. CSMA_PERSIST_EN = 1, Exit via API_STOP

7. Revise History

Table 29. Revise History

Version	Chapter	Revise Content	Date
0.5	All	Initial version	2020-09-17
0.6A	All	Review	2022-01-09
0.6B	6	Title name changed to "Carrier Sense Multiple Access (CSMA)" Added CSMA flowchart	2022-01-19
0.6B	3	Register name of CTL_REG_14 is changed to CTL_REG_22（0*16）in both Table 19 and Table 22.The RX_HOP_PERSIST corresponds to bit 6 and FREQ_SW_STATE corresponds to bit 5.	2022-04-26
0.7	All	Review	2022-08-12

CMOSTEK Confidential

8. Contacts

CMOSTEK Microelectronics Co., Ltd. Shenzhen Branch

Address: 30th floor, 8th Building, C Zone, Vanke Cloud City, Xili Sub-district, Nanshan, Shenzhen, GD, P.R. China

Post code: 518055

Tel： +86-755-83231427

Sales： sales@cmostek.com

Support： support@cmostek.com

Website： www.cmostek.com

CMOSTEK Confidential

All rights reserved at CMOSTEK Microelectronics Co., Ltd

The information furnished by CMOSTEK is believed to be accurate and reliable. However, CMOSTEK reserves rights to change, correct or modify products or documents at anytime without any notice. Without the written permission of the Company, no unit or individual shall copy part of or all of the contents of this document, and shall not disseminate it in any form. The content of this document will be updated irregularly due to product version upgrades or other reasons. CMOSTEK's products are not recommended to be applied to life-related equipment and systems, and CMOSTEK shall not be liable for any losses caused by the equipment or operation failure of the system in using the device.

CMOSTEK and some other related CMOSTEK is registered trademark of CMOSTEK Microelectronics Co., Ltd. However, all other registered names mentioned above are the property of their respective owners.