AK801

Beacon SOC User Manual

V1.0.9

Contents

1 Introduction

The AK801 is a single chip 2.4GHz MCU with an embedded baseband protocol engine, suitable for low power wireless applications. The AK801 is designed for operation in the world wide ISM frequency band at 2.400-2.480GHz.

The radio front end uses GFSK modulation. It has user configurable parameters like frequency channel, output power. AK801 supports an air data rate of 1Mbps, you can operate it with minimum number of peripheral components.

1.1 Features

Features of the AK801 include:

- Radio
  - Worldwide 2.4GHz ISM band operation (2.400GHz ~ 2.480GHz)
  - Common RX and TX interface
  - GFSK modulation
  - 1Mbps air data rate
  - 64 Bytes(Double-Buffer) or 128 Bytes(Single-Buffer) data length
  - CRC
  - Configurable whitening Filter
- Transmitter
  - 19mA at 0dBm output power
  - Programmable output power: -30 ~ +8 dBm
- Receiver
  - 18mA at 1Mbps
  - -95dBm at 1Mbps
- Power Management
  - 2.1 to 3.6V supply range
  - 2uA sleep-down mode
  - 5ms wake-up from sleep-down mode
  - 5ms start-up from power on
  - 250us sleep-down from standby mode
- Peripheral Interfaces
  - ADC (10-bit)
  - PWM
  - LVD
  - WDT
  - GPIO
  - I2C
- MCU
  - Crystal oscillator: 32MHz, 10PF, ±50ppm
  - OTP: 4K x 32 Bits

- RAM: 2K Bytes
- Basic Frequency: 16M

1.2 Packages

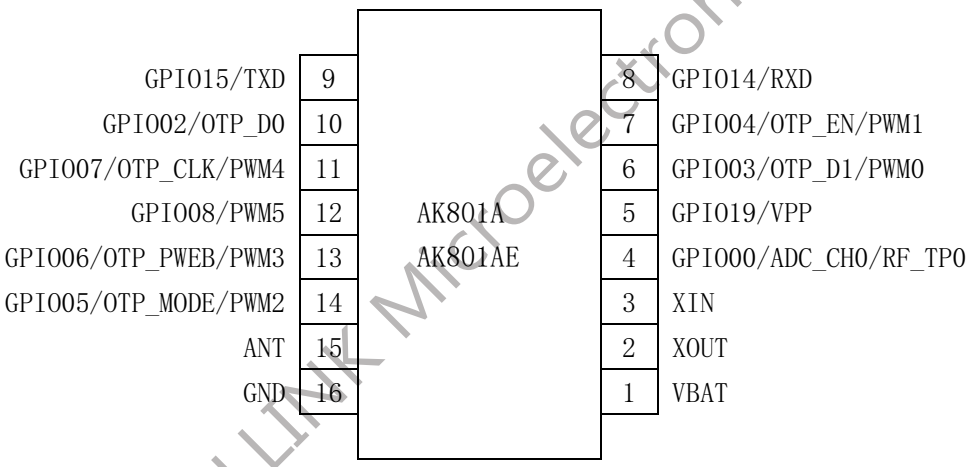
packages of the AK801 include:

ID	CHIP NAME	PACKAGE/GPIO_NUM/PWM_NUM	Features
1	AK801A	SOP16/11/6	SOP16 Basic
2	AK801AE	SOP16/11/6, internal E2PROM	SOP16 Basic + EEPROM
5	AK801E	SSOP24-patch 1.0	

2 Pin Information

2.1 SOP16

2.1.1 Pin assignment



2.1.2 Pin functions

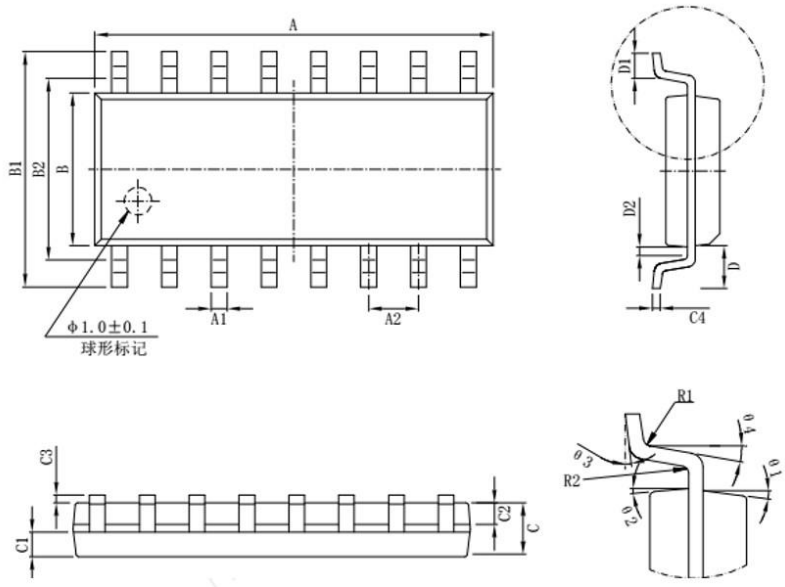
No.	Name	Type	Description
1	VBAT	POWER	Analog LDO Input
2	XOUT	OSC	32M OSC out
3	XIN	OSC	32M OSC in
4	GPIO00/ADC_CH0/RF_TP0	A/D	ADC CH0/GPIO00
5	GPIO19/VPP	IO/PWR	GPIO19/Open Drain, bonding with VPP6.5V
6	GPIO03/OTP_D1/PWM0	IO	
7	GPIO04/OTP_EN/PWM1	IO	
8	GPIO14/RXD	IO	
9	GPIO15/TXD	IO	
10	GPIO02/OTP_D0	IO	
11	GPIO07/OTP_CLK/PWM4	IO	

12	GPIO08/PWM5	IO	
13	GPIO06/OTP_PWEB/PWM3	IO	
14	GPIO05/OTP_MODE/PWM2	IO	
15	ANT	RF	ANT
16	GND	GND	VSS for ANT

2.1.3 Package size

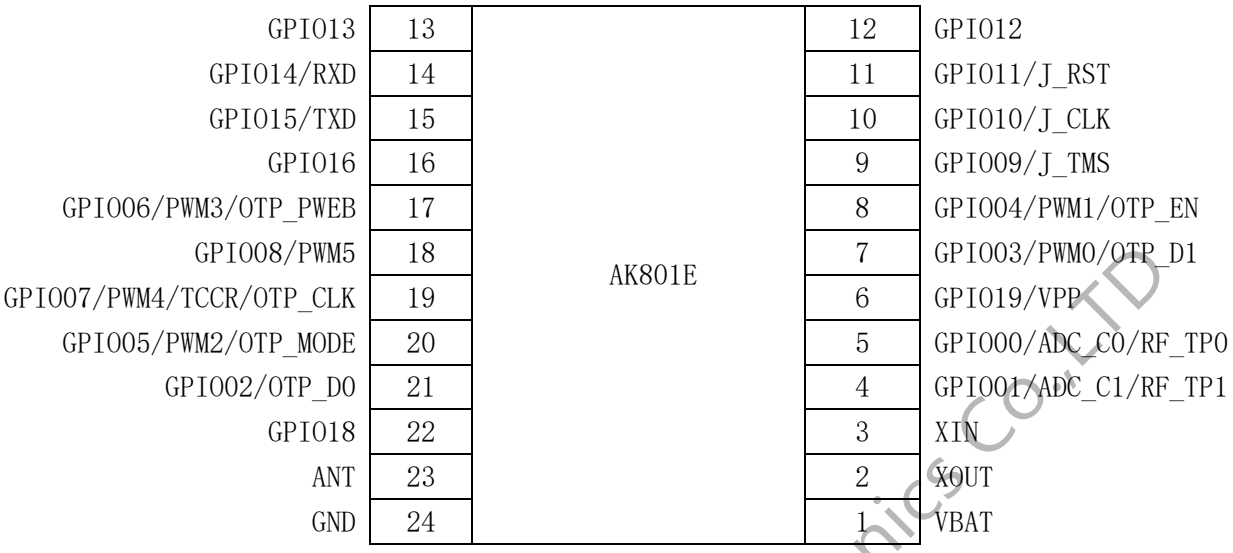
SOP16 封装尺寸

标注 / 尺寸	最小值（mm） MIN	最大值（mm） MAX
A	9.80	10.20
A1	0.35	0.50
A2	1.27 TYP	
B	3.80	4.00
B1	5.84	6.24
B2	5.00 TYP	
C	1.40	1.60
C1	0.61	0.71
C2	0.54	0.64
C3	0.05	0.25
C4	0.203	0.233
D	1.05 TYP	
D1	0.40	0.70
D2	0.15	0.25
R1	0.20 TYP	
R2	0.20 TYP	
Θ 1	8° ~ 12° TYP4	
Θ 2	8° ~ 12° TYP4	
Θ 3	0° ~ 8°	
Θ 4	4° ~ 12°	



2.2 SSOP24

2.2.1 Pin assignment



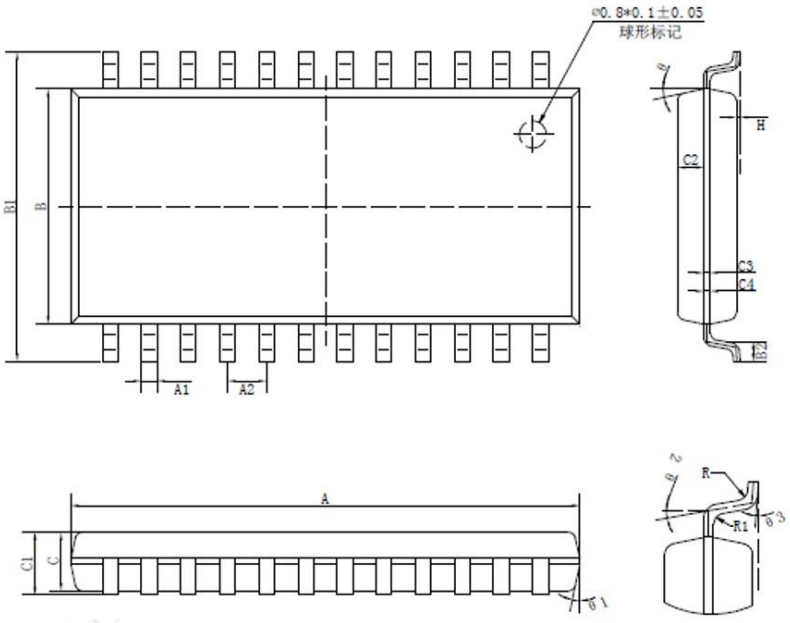
2.2.2 Pin functions

No.	Name	Type	Description
1	VBAT	POWER	ANALOG LDO INPUT
2	XOUT	OSC	32M OSC OUT
3	XIN	OSC	32M OSC IN
4	GPIO01/ADC_C1/RF_TP1	A/D	ADC CH1/GPIO01
5	GPIO00/ADC_C0/RF_TP0	A/D	ADC CH0/GPIO00
6	GPIO19/VPP	IO/PWR	GPIO19/OPENDRAIN,BONDING WITH VPP6.5V
7	GPIO03/PWM0/OTP_D1	IO	IO
8	GPIO04/PWM1/OTP_EN	IO	IO
9	GPIO09/J_TMS	IO	IO
10	GPIO10/J_CLK	IO	IO
11	GPIO11/J_RST	IO	IO
12	GPIO12	IO	IO
13	GPIO13	IO	IO
14	GPIO14/RXD	IO	IO
15	GPIO15/TXD	IO	IO
16	GPIO16	IO	IO
17	GPIO06/PWM3/OTP_PWEB	IO	IO
18	GPIO08/PWM5	IO	IO
19	GPIO07/PWM4/TCCR/OTP_CLK	IO	IO
20	GPIO05/PWM2/OTP_MODE	IO	IO
21	GPIO02/OTP_D0	IO	IO

22	GPIO18	IO	IO
23	ANT	RF	ANT
24	GND	GND	VSS FOR ANT

2.2.3 Package size

标注 / 尺寸	最小值（mm） MIN	最大值（mm） MAX
A	12.95	13.25
A1	0.35	0.50
A2	1.00 TYP	
B	5.95	6.05
B1	7.75	8.05
B2	0.40	0.60
C	1.45	1.55
C1	1.45	1.75
C2	0.674	
C3	0.152	
C4	0.172	
H	0.05	0.15
R	0.15 TYP	
R1	0.15 TYP	
Θ	12° TYP4	
Θ 1	12° TYP4	
Θ 2	10° TYP4	
Θ 3	0° ~ 8°	



3 Key Features

3.1 Electrical

Symbol	Parameter	Min	Typ	Max	Unit	Comment
VDD3	Supply Voltage, working mode	2.1	3.3	+3.6	V	
VSS					V	
VIO		-0.3		VDD+0.3	V	
ESD-HBM				2	kV	
ESD-MM				200	V	

3.2 RF

Name	Parameter	Min	Typ	Max	Unit	Comment
Normal RF Condition						
FOP	Operating frequency	2400		2480	MHz	
FXTAL	Crystal Frequency		32		MHz	
Transmitter Characteristics						
PRF	RF output power	-18		9	dBm	
PRF1	Out of band emission 2 MHz(GFSK)		-39		dBm	
PRF2	Out of band emission 3 MHz(GFSK)		-48		dBm	
BW	20dB bandwidth		1.1		MHz	BLE
Receiver Characteristics (1-Mbps GFSK (Bluetooth low energy))						
SEN	High Gain mode, Sensitivity @0.1%		-95		dBm	
FET	Frequency Error Tolerance	-300		300	kHz	
MaxIn	Maximum input power		-6		dBm	
C/I CO	Co-channel C/I, GFSK		7		dB	(1)
C/I1ST	ACS C/I 1MHz, GFSK		-4.5		dB	(2)
C/I1ST	ACS C/I -1MHz, GFSK		-4.5		dB	(3)
C/I2ND	ACS C/I 2MHz, GFSK		-35		dB	(4)
C/I2ND	ACS C/I -2MHz, GFSK		-17.5		dB	(5)
C/I3RD	ACS C/I 3MHz, GFSK		-39		dB	(6)
C/I3RD	ACS C/I -3MHz, GFSK		-37		dB	(7)
C/I Image	C/I Image frequency, GFSK		-17.5		dB	(8)

(1) Wanted signal at –67 dBm, modulated interferer in Channel, BER = 1E-3

(2) Wanted signal at –67 dBm, modulated interferer at +1 MHz, BER = 1E-3

(3) Wanted signal at –67 dBm, modulated interferer at -1 MHz, BER = 1E-3

(4) Wanted signal at –67 dBm, modulated interferer at +2 MHz, BER = 1E-3

(5) Wanted signal at –67 dBm, modulated interferer at -2 MHz, BER = 1E-3

(6) Wanted signal at –67 dBm, modulated interferer at +3 MHz, BER = 1E-3

(7) Wanted signal at –67 dBm, modulated interferer at -3 MHz, BER = 1E-3

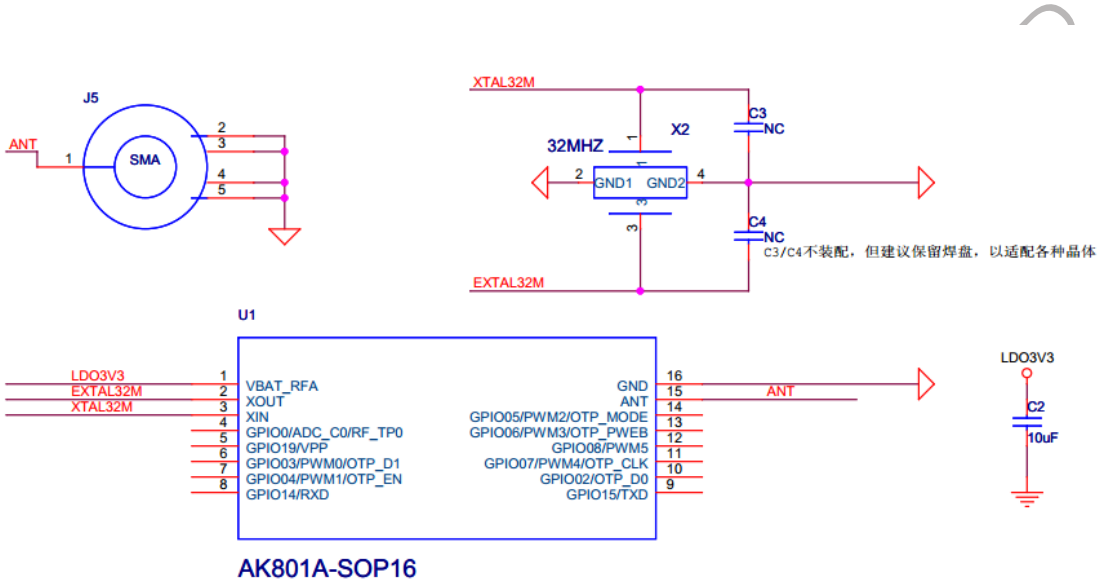
(8) Wanted signal at –67 dBm, modulated interferer at image frequency, BER = 1E-3

3.3 Temperature

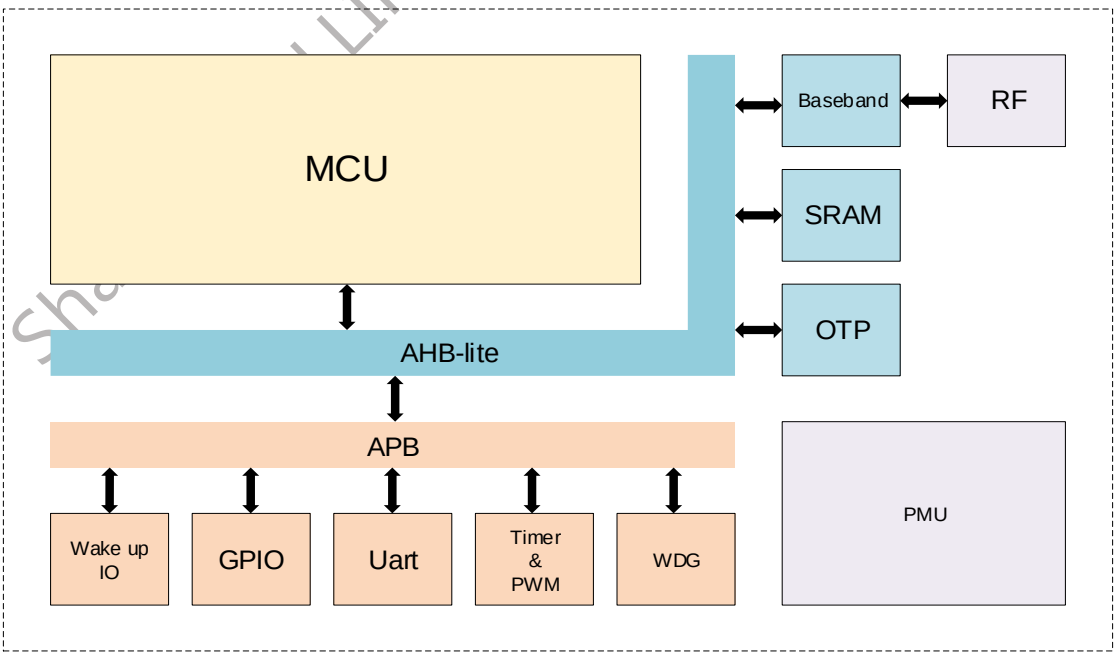
Chip type*	MIN	Typ.	MAX	Unit
H1	-20	20	85	℃
H3	-40	20	105	℃

** represents different chip types.*

4 Application Example



5 Block Diagram



6 Register Map

6.1 Soft-Reset Register

You can reset the MCU by software.

Register base address: 0x40120100

Soft-Reset register:

Register name	top_soft_reset_1		
Access type	RW_REG		
Offset	0x0C		
Reset value	0x00		
Description	Soft-Reset		
mapping	Domain name	Description	Reset value
[31:8]		reserverd	0x0
[7]	wdg_soft_rst	wdg software reset, high valid	0x0
[6]	timerccr_soft_rst	timer ccr software reset, high valid	0x0
[5]	bb_soft_rst	rfbb software reset, high valid	0x0
[4]	uart_soft_rst	uart software reset, high valid	0x0
[3]	gpio_soft_rst	gpio software reset, high valid	0x0
[2]	timer1_soft_rst	pwm timer1 software reset, high valid	0x0
[1]	timer2_soft_rst	pwm timer2 software reset, high valid	0x0
[0]	timer3_soft_rst	pwm timer3 software reset, high valid	0x0

Register name	top_soft_reset_2		
Access type	RW_REG		
Offset	0x10		
Reset value	0x00		
Description	soft_reset		
mapping	Domain name	Description	Reset value
[31:2]		reserverd	0x0
[1]	gpadc_soft_rst	gpadc software reset, high valid	0x0
[0]	wt_soft_rst	wt software reset, high valid	0x0

6.2 Clock Register

You can control the AK801 internal module clock separately.

Register name	top_clk_enable
Access type	RW_REG
Offset	0x14
Reset value	0x00
Description	clk_enable

mapping	Domain name	Description	Reset value
[31:7]		reserverd	0x0
[6]	timerccr_clk_en	timer_ccr clk enable, high valid	0x0
[5]	bb_clk_en	baseband clk enable, high valid	0x0
[4]	uart_clk_en	uart clk enable, high valid	0x0
[3]	gpio_clk_en	gpio clk enable, high valid	0x0
[2]	timer1_clk_en	pwm timer1 clk enable, high valid	0x0
[1]	timer2_clk_en	pwm timer2 clk enable, high valid	0x0
[0]	timer3_clk_en	pwm timer3 clk enable, high valid	0x0

6.3 Power Control Register

The register will be set to 0 if MCU resets. When system needs to be in sleep mode, config the register to 1. If MCU wakes up from sleep mode, the register will still be 1. When the MCU wakes up, you need to configure the register to 0.

Register name	power_ctrl_reg		
Access type	RW_REG		
Offset	0x04		
Reset value	0x00		
Description	BLE AGC config_1		
mapping	Domain name	Description	Reset value
[31:1]		reserverd	0x0
[0]	cpu_power_down	cpu control system power down, high valid, hard reset to set to 0.	0x0

6.4 Wakeup Source Register

Register name	wakeup_source		
Access type	RO_REG		
Offset	0x90		
Reset value	0x00		
Description	indicate wake up source		
mapping	Domain name	Description	Reset value
[31:3]		reserverd	0x0
[2:0]	wakeup_source	[0]:lvd wake up [1]:watch dog wake up [2]:wake up timer wake up	0x0

6.5 Interrupt Control Register

Address	Name	Permission	Initial Value	Description
0xE000E100	VIC_ISER	W/R	0x00000000	Interrupt-Enablement Setting
0xE000E140	VIC_IWER	W/R	0x00000000	Low-power Wake-up Setting

0xE000E180	VIC_ICER	W/R	0x00000000	Clear Interrupt-Enablement
0xE000E1C0	VIC_IWDR	W/R	0x00000000	Clear Low-power Wake-up
0xE000E200	VIC_ISPR	W/R	0x00000000	Interrupt-wait Setting
0xE000E280	VIC_ICPR	W/R	0x00000000	Clear Interrupt-wait
0xE000E300	VIC_IABR	W/R	0x00000000	Interrupt-response Status
0xE000E400- 0xE000E41C	VIC_IPR0- VIC_IPR7	W/R	0x00000000	Interrupt-priority Setting
0xE000EC00	VIC_ISR	R	0x00000000	Interrupt Status
0xE000EC04	VIC_IPTR	W/R	0x00000000	Interrupt priority threshold

6.5.1 Interrupt-Enablement Setting (VIC_ISER)

VIC_ISER is used to enable individual interrupts and to provide feedback on the enabling state of each interrupt.

BIT	NAME	Description
31:0	SETENA	Set to use to read the enabled state of one or more interrupts. Each bit corresponds to the same number of interrupt source: Read operation: ➢ 0: The corresponding interrupt is not enabled. ➢ 1: The corresponding interrupt is enabled. Write operation ➢ 0: invalid. ➢ 1: Enable the corresponding interrupt.

If a waiting interrupt is enabled, the Vector Interrupt Controller activates the interrupt based on its priority. If an interrupt is not enabled, the vector interrupt controller will not activate the interrupt even if the interrupt is in the wait state.

6.5.2 Low-power Wake-up Setting (VIC_IWER)

VIC_IWER is used to enable low-power wake up for individual interrupts and to feedback the enabling state of low-power wake up for individual interrupts.

BIT	NAME	Description
31:0	SETENA	Set to use to read one or more enabled states. Each bit corresponds to the same number of interrupt source: Read operation: ➢ 0: The corresponding function is not enabled. ➢ 1: The corresponding function is enabled. Write operation ➢ 0: invalid. ➢ 1: Enable the corresponding function (low-power wake-up).

If a low-power wake-up function for an interrupt is enabled and the interrupt is in a wait state, the VIC generates a low-power wake up request. If an interrupt's low-power wake up function is not

enabled, the VIC does not generate low-power wake up requests even if the interrupt is in the wait state.

Note: interrupt enablement and interrupt wake enablement control interrupt transaction and interrupt wake, respectively. When both are set, an interrupt in the waiting state generates both an interrupt request and a low-power wake request. When only one of them is enabled, only the corresponding function is activated; When neither is enabled, no interrupt request or low-power wake request is generated even if the interrupt is in the wait state.

6.5.3 Clear Interrupt-Enablement

VIC_ICER is used to clear the enabling of individual interrupts and to feed back the enabling status of each interrupt.

BIT	NAME	Description
31:0	CLRENA	Clear interrupt-Enablement. Read the enabled state of one or more interrupts. Each bit corresponds to the same number of interrupt source: Read operation: ➢ 0: The corresponding interrupt is not enabled. ➢ 1: The corresponding interrupt is enabled. Write operation ➢ 0: invalid. ➢ 1: Clear the corresponding interrupt-enablement.

6.5.4 Clear Low-power Wake-up-enablement（VIC_IWDR）

VIC_IWDR is used to clear low-power wake-up-enablement for individual interrupts and to feed back the enabling state of low-power wake-ups for each interrupt.

BIT	NAME	Description
31:0	IWDR	Clear low-power wake-up-enablement. Read the enabled state of one or more interrupts. Each bit corresponds to the same number of interrupt source: Read operation: ➢ 0: The corresponding function of the interrupt is not enabled. ➢ 1: The corresponding function of the interrupt is enabled. Write operation ➢ 0: invalid. ➢ 1: Clear the corresponding low-power wake-up-enablement.

6.5.5 Interrupt-wait Setting（VIC_ISPR）

VIC ISPR means to set the interrupt to wait state.

BIT	NAME	Description
31:0	SETPEND	Change one or more interrupts to the wait state. Each bit corresponds to the same number of interrupt source: Read operation: ➢ 0: The corresponding interrupt is not in the wait state. ➢ 1: The corresponding interrupt is in the wait state. Write operation ➢ 0: invalid. ➢ 1: Set the corresponding interrupt to the wait state.

6.5.6 Clear Interrupt-wait (VIC_ICPR)

VIC_ICPR means to clear the waiting state of each interrupt.

BIT	NAME	Description
31:0	CLRPEND	Clear one or more interrupts to the wait state. Each bit corresponds to the same number of interrupt source: Read operation: ➢ 0: The corresponding interrupt is not in the wait state. ➢ 1: The corresponding interrupt is in the wait state. Write operation ➢ 0: invalid. 1: Clear the wait state of the corresponding interrupt.

6.5.7 Interrupt-response Status（VIC_IABR）

VIC_IABR is used to indicate the current ACTIVE status of each interrupt and is a register for query. In addition, the software can clear the ACTIVE status of all interrupts to 0 when initializing VIC.

BIT	NAME	Description
31:0	CLRPEND	Query bits indicating whether the interrupt source has been responded by the CPU but has not yet been processed. Each bit corresponds to the same number of interrupt sources. Read operation: ➢ 0: The CPU doesn't response it. ➢ 1: The CPU has responded to the interrupt but has not finished processing it. Write operation: ➢ 0: Clear the Active state of interrupt. ➢ 1: can't write 1.

6.5.8 Interrupt-priority Setting (VIC_IPR0 - VIC_IPR7)

Each interrupt priority setting register provides priority Settings for four interrupt sources. Depending on the application definition, each interrupt source setting area corresponds to a corresponding interrupt source. For a hardware implementation that supports 32 interrupt sources, registers from IPR0 to IPR7 are shown in the diagram.

The vector interrupt controller selects the interrupt priority according to the priority number. The smaller the priority number, the higher the priority. If the priority number is the same, the interrupt source number determines the priority order; the smaller the number, the higher the priority.

	31 24	23 16	15 8	7 0
IPR0	PRI_3	PRI_2	PRI_1	PRI_0
IPRn	PRI_4n+3	PRI_4n+2	PRI_4n+1	PRI_4n
IPR7	PRI_31	PRI_30	PRI_29	PRI_28

31 30	29 24	23 22	21 16	15 14	13 8	7 6	5 0
PRI_N+3	Reserved	PRI_N+2	Reserved	PRI_N+1	Reserved	PRI_N	Reserved

interrupt priority setting register

The following table describes the bit definitions for VIC_IPRn. In this table, N = 4n, n is the VIC_IPRN register number. For example, VIC_IPR2, n is 2, N is 8.

BIT	NAME	Description
31:30	PRI_N+3	The interrupt number is N+3. The lower the value, the higher the priority.
29:24		Reserved
23:22	PRI_N+2	The interrupt number is N+2. The lower the value, the higher the priority.
21:16	-	Reserved
15:14	PRI_N+1	The interrupt number is N+1. The lower the value, the higher the priority.
13:8	-	Reserved
7:6	PRI_N	The interrupt number is N. The lower the value, the higher the priority.
5:0	-	Reserved

6.5.9 Interrupt Status (VIC_ISR)

VIC_ISR indicates the interrupt vector number currently being processed by the CPU and the interrupt vector number with the highest priority waiting. This register is a read-only register for software queries.

BIT	NAME	Description
31:21	Reserved	Reserved
20:12	VECTPENDING	Indicates the interrupt vector number with the highest priority that is currently in the wait state;
11:9	Reserved	Reserved
8:0	VECTACTIVE	Indicates the interrupt vector number that the CPU is currently processing.

6.5.10 Interrupt priority threshold (VIC_IPTR)

VIC_IPTR defines the priority threshold at which an interrupt request currently in the waiting state can initiate interrupt preemption. The priority of the waiting interrupt request must be higher than the priority threshold defined by VIC_IPTR in order to initiate the interrupt preemption request.

BIT	NAME	Description
31	EN	Interrupt priority threshold valid bits: ➢ 0: interrupt preemption does not need a priority higher than the threshold ➢ 1: Interrupt preemption requires a priority above the threshold
30:17	Reserved	Reserved
16:8	VECTTHRESHOLD	Indicates the interrupt vector number corresponding to the priority threshold. When the VIC finds that the CPU exits from the interrupt service routine corresponding to the VECTTHRESHOLD, the hardware automatically clears the significant bits of the interrupt priority threshold.
7:0	PRIOTHRESHOLD	Indicates a priority threshold for interrupt preemption. Note: In CK801, only 2 higher bits [7:6] are valid, and the rest [5:0] are reserved bits.

6.6 GPIO Register

6.6.1 Multiplexing functional register

Base Address：0x40120000

Name	PAD_GPIOX		
Access Type	RW_REG		
Offset	0x0 – 0x4C		
Reset Value	0x01		
Description			
Mapping	Domain	Description	Reset value
[31:5]		reserved	0x0
[4:2]	PAD_GPIOXX_FUNC	Mux function: Padmux function index	0x0

[1]	PAD_GPIOXX_IE	1: Receive enable/output loop back 0:	0x0
[0]	PAD_GPIOXX_REN	1: Do not pull up 0: Enable pull-up	0x1

Padmux function index	Value
PAD_MUX_FUNCTION_0	0x00
PAD_MUX_FUNCTION_1	0x04
PAD_MUX_FUNCTION_2	0x08
PAD_MUX_FUNCTION_3	0x0c
PAD_MUX_FUNCTION_4	0x10
PAD_MUX_FUNCTION_5	0x14
PAD_MUX_FUNCTION_6	0x18
PAD_MUX_FUNCTION_7	0x1c

GPIO_BASE_ADDR 0x40060000

6.6.2 Output data register

Name	GPIO_SWPORT_A_DR		
Access Type	RW_REG		
Offset	0x00		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset value
[31:20]		reserved	0x0
[19:0]	GPIO_SWPORT_A_DR	1: High-level 0: Low-level	0x0

6.6.3 configuration register

Name	GPIO_SWPORT_A_DDR		
Access Type	RW_REG		
Offset	0x04		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset value
[31:20]		reserved	0x0
[19:0]	GPIO_SWPORT_A_DDR	0: input(default) 1: output	0x0

6.6.4 External port register

Name	GPIO_EXT_PORT_A		
Access Type	RW_REG		
Offset	0x50		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset value
[31:20]		reserved	0x0
[19:0]	GPIO_SWPORT_A_DDR	0: low level 1: high level	0x0

6.6.5 Interrupt Enable Register

Name	GPIO_INT_EN		
Access Type	RW_REG		
Offset	0x30		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset value
[31:20]		reserved	0x0
[19:0]	GPIO_INT_EN	0: disable interrupt 1: enable interrupt	0x0

6.6.6 interrupt mask register

Name	GPIO_INT_MASK		
Access Type	RW_REG		
Offset	0x34		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset value
[31:20]		reserved	0x0
[19:0]	GPIO_INT_MASK	0: Unmasked interrupt 1: Masked interrupt	0x0

6.6.7 Interrupt type register

Name	GPIO_INT_TYPE_LEVEL
Access Type	RW_REG
Offset	0x38
Reset Value	0x0
Description	

Mapping	Domain	Description	Reset value
[31:20]		reserved	0x0
[19:0]	GPIO_INT_TYPE_LEVEL	0: Level interrupt 1: Interruption of the edge	0x0

6.6.8 Interrupt type register

Name	GPIO_INT_POLARITY		
Access Type	RW_REG		
Offset	0x3c		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset value
[31:20]		reserved	0x0
[19:0]	GPIO_INT_POLARITY	0: Low level/falling edge 1: High level/rising edge	0x0

6.6.9 Interrupt status register

Name	GPIO_INT_STATUS		
Access Type	RW_REG		
Offset	0x40		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset Value
[31:20]		reserved	0x0
[19:0]	GPIO_INT_STATUS	0: Reset Value 1: Interrupt flag	0x0

6.6.10 Raw interrupt status register

Name	GPIO_RAW_INT_STATUS		
Access Type	RW_REG		
Offset	0x44		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset Value
[31:20]		reserved	0x0
[19:0]	GPIO_INT_STATUS	0: Reset Value 1: Original interrupt flag	0x0

6.6.11 Dithering register

Name	GPIO_DEBOUNCE

Access Type	RW_REG		
Offset	0x48		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset Value
[31:20]		reserved	0x0
[19:0]	GPIO_DEBOUNCE	0: no debounce 1: open the debounce	0x0

6.6.12 Interrupt status-clear register

Name	GPIO_PORT_A_EOI		
Access Type	RW_REG		
Offset	0x48		
Reset Value	0x0		
Description			
Mapping	Domain	Description	Reset Value
[31:20]		reserved	0x0
[19:0]	GPIO_PORT_A_EOI	1: clear the interrupt flag	0x0

6.6.13 Top control register

Address：0x40120100

Name	top_ctrl_reg		
Access Type	RW_REG		
Offset	0x00		
Reset Value	0x50		
Description	top_ctrl config		
Mapping	Domain	Description	Reset Value
[31:8]		reserverd	0x0
[7]	start_gp_adc	gpadc ctrl signal, high vaild	0x0
[6:5]	pin0_mode_ull	ADC_CH0 pin configure 00: For GPADC 01: For ABB Test 10: For Digital input from pin 11: For Digital output to pin	0x2
[4:3]	pin1_mode_ull	ADC_CH1 pin configure 00: For GPADC 01: For ABB Test 10: For Digital input from pin 11: For Digital output to pin	0x2

[2]	en_gpadc_ull	Enable general purpose ADC, active high	0x0
[1]	lvd_pwr_ctrl_en	enable lvd power up, high vaild	0x0
[0]	pmu_ctrl_enable	enable power ctrl state to control power flow, 1: enable, 0: use top register to control power flow	0x0

6.7 TimerCCR Register

Base Address：0x4001_0000

NAME	WIDTH	ADDR	TYPE	DESCRIPTION
TCCR_EN	1	00[0]	R/W	TimerCCR work enable, high is assert
EX_RELOAD_EN	1	00[1]	R/W	0: external reload disable 1: external reload is enabled（When tccr_ex falledge）
TIMER_RELOAD_SEL	2	00[3:2]	R/W	TimerCCR reload mode selection: 0: Disable reload mode, counter on free-run; 1: Software force reload mode, when set, reload counter immediately and counter is stopped at this time; 2: auto-reload upon counter overflow; 3: reload at falling edge of input pin tccr_ex;
TIMER_CLK_SEL	2	00[5:4]	R/W	TimerCCR clock selection: 0: timerCCR is stopped; 1: normal mode, counter clock is clk/2; 2: Counter add at tccr falling edge 3: Counter only add when tccr is high as clk/2;
TCCR_INV	1	00[6]	R/W	tccr_ex inv enable
TIMER_RELOAD	24	04[23:0]	R/W	Timer reload value
CC0_C	24	18[23:0]	RO	Capture 0 value timer counter value at the posedge of tccr_ex
CC1_C	24	1C[23:0]	RO	Capture 1 value timer counter value at the posedge of tccr_ex
TIMERCCR_INT_EN	2	28[1:0]	R/W	Interrupt enable {tccr_ov, tccr_ex_negedge}
TIMERCCR_INT_MSK	2	2C[1:0]	R/W	Interrupt Mask, when 1, no interrupt

				present, but status register no affect. {tccr_ov, tccr_ex_negedge}
TIMERCCR_INT_CLR	2	30[1:0]	R/W	Write 1 to clear, and reading from this register returns zeros(0). {tccr_ov, tccr_ex_negedge}
TIMERCCR_INT_STS	2	34[1:0]	RO	Interrupt status {tccr_ov, tccr_ex_negedge}

6.8 Timer0 Register

Base Address: 0x4003_0000

Address	Function
0x00	Timer0LoadCount
0x04	Timer0CurrentValue
0x08	Timer0ControlReg
0x0C	Timer0EOI
0x10	Timer0IntStatus
0xA0	TimersIntStatus
0xA4	TimersEOI
0xA8	TimersRawIntStatus

6.8.1 Timer0LoadCount

Name	Timer0LoadCount		
Access Type	RW_REG		
Offset	0x00		
Reset value	0x0000_0000		
Description	Reload value of TIMER0		
Mapping	Domain	Description	Reset Value
[31:17]		reserved	0x0
[16:0]		Reload value of TIMER0.	0x0

6.8.2 Timer0CurrentValue

Name	Timer0CurrentValue		
Access Type	R_REG		
Offset	0x04		
Reset value	0x0000_0000		
Description	Current value of TIMER0		
Mapping	Domain	Description	Reset value
[31:16]		reserved	0x0
[15:0]		Current value of TIMER0	0x0

6.8.3 Timer0ControlReg

Name	Timer0ControlReg		
Access Type	RW_REG		
Offset	0x08		
Reset value	0x0000_0000		
Description	Timer0 Control register		
Mapping	Domain	Description	Reset value
[31:3]		reserved	0x0
[2]		The interrupt mask of Timer0, High is valid.	0x0
[1]		1：Reload the value configured. 0：Reload all 1 only.	
[0]		1: Timer0 is enable	

6.8.4 Timer0EOI

Name	Timer0EOI		
Access Type	RC_REG		
Offset	0x0C		
Reset value	0x0000_0000		
Description	Clear timer0 interrupt by reading.		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]		Clear timer0 interrupt by reading, result value is 0.	0x0

6.8.5 Timer0IntStatus

Name	Timer0IntStatus		
Access Type	R_REG		
Offset	0x10		
Reset value	0x0000_0000		
Description	The interrupt status of Timer0		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]		The interrupt status of Timer0	0x0

6.8.6 TimersIntStatus

Name	TimersIntStatus
Access Type	R_REG
Offset	0xA0

Reset value	0x0000_0000		
Description	Interrupt status		
Mapping	Domain	Description	Reset value
[31:3]		reserved	0x0
[2:0]		Interrupt status bit[0]: timer0 bit[1]: timer1 bit[2]: timer2	0x0

6.8.7 TimersEOI

Name	TimersEOI		
Access Type	RC_REG		
Offset	0xA4		
Reset value	0x0000_0000		
Description	Clear timer_x interrupt by reading.		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]		Clear timer_x interrupt by reading.	0x0

6.8.8 TimersRawIntStatus

Name	TimersRawIntStatus		
Access Type	R_REG		
Offset	0xA8		
Reset value	0x0000_0000		
Description	The status of interrupt source		
Mapping	Domain	Description	Reset value
[31:3]		reserved	0x0
[2:0]		The interrupt source state, without the interrupt information processed by the mask. bit[0]: timer1 bit[1]: timer2 bit[2]: timer3	0x0

6.9 Timer1 Register

Base Address: 0x4003_0000

Address	Function
0x14	Timer1LoadCount
0x18	Timer1CurrentValue

0x1C	Timer1ControlReg
0x20	Timer1EOI
0x24	Timer1IntStatus
0xA0	TimersIntStatus
0xA4	TimersEOI
0xA8	TimersRawIntStatus

6.9.1 Timer1LoadCount

Name	Timer1LoadCount		
Access Type	RW_REG		
Offset	0x14		
Reset value	0x0000_0000		
Description	The reload value of Timer1.		
Mapping	Domain	Description	Reset value
[31:16]		reserved	0x0
[15:0]		Reload value of Timer1, the period of timer1 is (reload_value+1).	0x0

6.9.2 Timer1CurrentValue

Name	Timer1CurrentValue		
Access Type	R_REG		
Offset	0x18		
Reset value	0x0000_0000		
Description	The current value of Timer1		
Mapping	Domain	Description	Reset value
[31:16]		reserved	0x0
[15:0]		The current value of Timer1.	0x0

6.9.3 Timer1ControlReg

Name	Timer1ControlReg
Access Type	RW_REG
Offset	0x1C
Reset value	0x0000_0000

Description	The control-register of Timer1		
Mapping	Domain	Description	Reset value
[31:3]		reserved	0x0
[2]		The interrupt mask of Timer1, high is valid.	0x0
[1]		1：Reload the value configured. 0：Reload all 1 only.	
[0]		1: Timer1 is enable.	

6.9.4 Timer1EOI

Name	Timer1EOI		
Access Type	RC_REG		
Offset	0x20		
Reset value	0x0000_0000		
Description	Clear timer1 interrupt by reading.		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]		Clear timer1 interrupt by reading, result value is 0.	0x0

6.9.5 Timer1IntStatus

Name	Timer1IntStatus		
Access Type	R_REG		
Offset	0x24		
Reset value	0x0000_0000		
Description	The interrupt status of Timer1		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]		The interrupt status of Timer1	0x0

Note: Timer1 source interrupt status and interrupt status are reflected in registers TimersIntStatus and TimerRawIntStatus in the Timer0 section. Reading registers TimersEOI in the Timer0 section will clear the interrupt source register directly to zero.

6.10 Timer2 Register

Base Address：0x4003_0000

Address	Function
0x28	Timer3LoadCount
0x2C	Timer3CurrentValue

0x30	Timer3ControlReg
0x34	Timer3EOI
0x38	Timer3IntStatus
0xA0	TimersIntStatus
0xA4	TimersEOI
0xA8	TimersRawIntStatus

6.10.1 Timer2LoadCount

Name	Timer2LoadCount		
Access Type	RW_REG		
Offset	0x28		
Reset value	0x0000_0000		
Description	The reload count of Timer2		
Mapping	Domain	Description	Reset value
[31:16]		reserved	0x0
[15:0]		Reload value of Timer2, the period of timer2 is (reload_value+1).	0x0

6.10.2 Timer2CurrentValue

Name	Timer3CurrentValue		
Access Type	R_REG		
Offset	0x2C		
Reset value	0x0000_0000		
Description	The current value of Timer1		
Mapping	Domain	Description	Reset value
[31:16]		reserved	0x0
[15:0]		The current value of Timer1	0x0

6.10.3 Timer2ControlReg

Name	Timer3ControlReg		
Access Type	RW_REG		
Offset	0x30		
Reset value	0x0000_0000		
Description	The control-register of Timer2		
Mapping	Domain	Description	Reset value
[31:3]		reserved	0x0

[2]		The interrupt mask of Timer1, high is valid.	0x0
[1]		1：Reload the value configured. 0：Reload all 1 only.	
[0]		1: Timer1 is enable.	

6.10.4 Timer2EOI

Name	Timer2EOI		
Access Type	RC_REG		
Offset	0x34		
Reset value	0x0000_0000		
Description	Clear timer2 interrupt by reading.		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]		Clear timer2 interrupt by reading, result value is 0.	0x0

6.10.5 Timer2IntStatus

Name	Timer2IntStatus		
Access Type	R_REG		
Offset	0x38		
Reset value	0x0000_0000		
Description	The interrupt status of Timer2		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]		The interrupt status of Timer2	0x0

Note: Timer1 source interrupt status and interrupt status are reflected in registers TimersIntStatus and TimerRawIntStatus in the Timer0 section. Reading registers TimersEOI in the Timer0 section will clear the interrupt source register directly to zero.

6.11 CoreTimer Register

Address	Function
0xE000E010	CORET_CSR
0xE000E014	CORET_RVR
0xE000E018	CORET_CVR
0xE000E01C	CORET_CALIB

6.11.1 Control-Status Register (CORET_CSR)

Name	CORET_CSR		
Offset	0xE000E010		
Reset value	0x0000_0004		
Description	System timer control and status register		
Mapping	Type	Description	Reset value
[31:17]	-	reserved	0x0
[16]	Read only	Indicates whether the counter has counted to 0 since the last read of this register: ➢ 0: counter has not yet counted to 0 ➢ 1: counter has already counted to 0 CountFlag is set when the value of the counter changes from 1 to 0. Reading the CSR register and any writing to the CVR register clears CountFlag to zero.	0x0
[15:3]	-	reserved	0x0
[2]	W/R	Represents the clock source of the system timer： ➢ 0: uses an optional external reference clock as the clock source for the counter ➢ 1: Use the internal clock as the clock source for the counter If there is no external clock, reading this bit returns 1, and writing this bit has no effect. The external reference clock frequency must be less than or equal to half of the internal clock frequency.	0x0
[1]	W/R	Indicates whether the system timer triggers an interrupt when the count reaches 0: ➢ 0: The system timer does not trigger an interrupt when the count reaches 0 ➢ 1: Writing the CVR register clears the counter to zero, but this method does not cause the system timer to trigger an	0x0

		interrupt.	
[0]	W/R	Represents the enabled status bit of the system timer: ➢ 0: counter is not enabled ➢ 1: counter is enabled	

6.11.2 Reload value Register (CORET_RVR)

Name	CORET_RVR		
Offset	0xE000E014		
Reset value	0x0000_0000		
Description	The CORET_RVR register is used to assign a value to the CORET_CVR register at the beginning of each counting cycle.		
Mapping	Type	Description	Reset value
[31:24]	-	reserved	0x0
[23:0]	W/R	When the counter reaches 0, the RELOAD value is assigned to the CORET_CVR register. Writing a 0 to the CORET_RVR register causes the counter to stop working after the current count has been completed, and the value of the counter will remain zero thereafter. When using an external reference clock can make counter, RELOAD value by external clock assigned to CORET_CVR, so must be to ensure the success of the RELOAD assignment, counter count after the start of the normal (namely CORET_CVR into a non-zero value), CORET_RVR can be set to 0, so that the counter after the completion of the current count to stop working, or counter will not be able to start counting.	0x0

How do I calculate the reload value:

The normal value of Reload is between 0x1 and 0x00FFFFFF. To produce a timer with N counting clock cycles, the value of Reload needs to be assigned to n-1. For example, to generate a CoreTim interrupt every 100 clock cycles, you need to assign a value of 99 to Reload.

6.11.3 Current value register (CORET_CVR)

Name	CORET_RVR

Offset	0xE000E018		
Reset value	0x0000_0000		
Description	Current value of the core timer		
Mapping	Type	Description	Reset value
[31:24]	-	reserved	0x0
[23:0]	Read	It indicates the value of the counter when it is read. Writing the CORET_CVR register clears both the register and the CountFlag status bit. Further, it causes the system timer to fetch the value of the register CORET_RVR and assign it to CORET_CVR in the next clock cycle. Note that writing CORET_CVR does not interrupt the system timer, and reading CORET_CVR returns the value of the current counter.	0x0

6.11.4 Calibration register (CORET_CALIB)

The CORET_CALIBB register describes the calibration function of the system timer.Its Reset value depends on the implementation: you need to get more information on the meaning of the CORET_CALIBB bit from the documentation provided by the device vendor, and the check value TENMS in the CORET_CALIBB register. Based on the calibration value of TENMS, the software can multiply this value by a certain proportion to obtain other different count periods, which must be within the range of the counter.

Name	CORET_CALIB		
Offset	0xE000E01C		
Reset value	-		
Description	Current value of the core timer		
Mapping	Type	Description	Reset value
[31]	Read only	Indicates whether the device implements an external reference clock: ➢ 0: The device has an external reference clock ➢ 1: The device has no external reference clock When this is 1, the CLKSource bit of the CORET_CSR register is fixed to 1 and cannot be overwritten.	0x0
[30]	Read only	Denotes whether the 10ms calibration value is accurate:	0x0

		➢ 0: 10ms calibration value is accurate ➢ 1: 10ms calibration value is incorrect due to the clock frequency	
[29:24]	-	reserved	0x0
[23:0]	Read only	Used to represent the backfill value corresponding to the 10ms time. Depending on the exact value of the SKEW, it may indicate a completely accurate 10ms value or the nearest 10ms value. If the value of this field is 0, the calibration value is unknown. This may be because the reference clock is an unknown input or is dynamically changing.	0x0

Note: If CORET_RVR is set to 0, the CoreTim counter will stop working on the next turn, regardless of the counter's enable bit status.

The value of the SYST_CVR register is unknown at reset time. Before enabling the CoreTim counter, the software must first write the required count to the CORET_RVR register and then write any value to the CORET_CVR, which clears the CORET_CVR value to zero. After enabling the counter, the counter can read the CORET_RVR backfill register and count down from that value, avoiding the need to start counting from an arbitrary value.

Operation steps:
Since there is no Reset value in the CORET_RVR and CORET_CVR registers in the system timer, you must follow the following steps before the system timer works:
1) Write the required backfill value to the CORET_RVR register;
2) Write any value to the CORET_CVR register to clear it to zero;
3) Operate the CORET_CSR register to enable the system timer.

6.12 PWM Register

Base Address：0x4003_0000

Address	Function
0xB0	TIMER0_PWM_CTL
0xB4	TIMER0_PWM0_DUTY_CYCLE
0xB8	TIMER0_PWM1_DUTY_CYCLE

0xBC	TIMER0_PWM2_DUTY_CYCLE
0xC0	TIMER0_PWM3_DUTY_CYCLE
0xC4	TIMER0_PWM4_DUTY_CYCLE
0xC8	TIMER0_PWM5_DUTY_CYCLE
0xCC	TIMER0_PWM_EN
0xD0	TIMER0_PWM_FD
0xD4	TIMER0_PWM_BOUND_EN
0xD8	TIMER0_PWM_BOUND_VAL

6.12.1 TIMER0_PWM_CTL

Name	TIMER0_PWM_CTL		
Access Type	RW_REG		
Offset	0xB0		
Reset value	0x0000_0000		
Description	PWM Control		
Mapping	Domain	Description	Reset value
[31:12]		reserved	0x0
[11:6]	timer0_pwm_mode	PWM output mode ➢ 0: There will be no jump when the PWM signal is high. One PWM cycle contains one low level and one high level. ➢ 1: PWM will jump when it is high; Bit Description: [5] PWM[5] output mode [4] PWM[4] output mode [3] PWM[3] output mode [2] PWM[2] Output mode [1] PWM[1] output mode [0] PWM[0] Output Mode	0x0
[5:0]	timer0_pwm_cmp	PWM output compensation, high efficiency bit description [5] PWM[5] output compensation [4] PWM[4] output compensation [3] PWM[3] output compensation	0x0

		[2] PWM[2] output compensation [1] PWM[1] output compensation [0] PWM[0] output compensation	

6.12.2 TIMER0_PWMN_DUTY_CYCLE

Name	TIMER0_PWMN_DUTY_CYCLE		
Access Type	RW_REG		
Offset	for N=0, 0xB4 for N=1, 0xB8 for N=2, 0xBC for N=3, 0xC0 for N=4, 0xC4 for N=5, 0xC8		
Reset value	0x0000_0000		
Description	PWM duty cycle		
Mapping	Domain	Description	Reset value
[31:16]		reserved	0x0
[15:0]	timer0_pwmN_duty_cycle	When timer0 is equal to this set of register values, the PWM will be set to high.	0x0

6.12.3 TIMER0_PWM_EN

Name	TIMER0_PWM_EN		
Access Type	RW_REG		
Offset	0xCC		
Reset value	0x0000_0000		
Description	Enable PWM to output.		
Mapping	Domain	Description	Reset value
[31:6]		reserved	0x0
[5:0]	timer1_pwm_en	Enable PWM to output, high is effective. Bit Description： [5] PWM[5] Output Enable [4] PWM[4] Output Enable [3] PWM[3] Output Enable [2] PWM[2] Output Enable [1] PWM[1] Output Enable [0] PWM[0] Output Enable	0x0

6.12.4 TIMER0_PWM_FD

Name	TIMER0_PWM_FD		
Access Type	RW_REG		
Offset	0xD0		
Reset value	0x0000_0000		
Description	Settings of pwm period.		
Mapping	Domain	Description	Reset value
[31:12]		reserved	0x0
[11:10]	pwm5_fd_mode	Settings of pwm5 period. 2'h0: period is unchanged 2'h1: period/2 2'h2: period/4 2'h3: period/8	0x0
[9:8]	pwm4_fd_mode	Settings of pwm4 period. 2'h0: period is unchanged 2'h1: period/2 2'h2: period/4 2'h3: period/8	0x0
[7:6]	pwm3_fd_mode	Settings of pwm3 period. 2'h0: period is unchanged 2'h1: period/2 2'h2: period/4 2'h3: period/8	0x0
[5:4]	pwm2_fd_mode	Settings of pwm2 period. 2'h0: period is unchanged 2'h1: period/2 2'h2: period/4 2'h3: period/8	0x0
[3:2]	pwm1_fd_mode	Settings of pwm1 period. 2'h0: period is unchanged 2'h1: period/2 2'h2: period/4 2'h3: period/8	0x0
[1:0]	pwm0_fd_mode	Settings of pwm0 period. 2'h0: period is unchanged 2'h1: period/2 2'h2: period/4 2'h3: period/8	0x0

6.12.5 TIMER0_PWM_BOUND_EN

Name	TIMER0_PWM_BOUND_EN		
Access Type	RW_REG		
Offset	0xD4		
Reset value	0x0000_0000		
Description	Enable the PWM dead-zone		
Mapping	Domain	Description	Reset value
[31:6]		reserved	0x0
[5:0]	timer1_pwm_bound_en	Enable the PWM dead-zone, high is effective. Bit Description： [5] PWM[5] enable the dead-zone [4] PWM[4] enable the dead-zone [3] PWM[3] enable the dead-zone [2] PWM[2] enable the dead-zone [1] PWM[1] enable the dead-zone [0] PWM[0] enable the dead-zone	0x0

6.12.6 TIMER0_PWM_BOUND_VAL

Name	TIMER0_PWM_BOUND_VAL		
Access Type	RW_REG		
Offset	0xD4		
Reset value	0x0000_0000		
Description	Config the value of PWM dead-zone		
Mapping	Domain	Description	Reset value
[31:6]		reserved	0x0
[5:0]	timer1_pwm_bound_val	Config the value of PWM dead-zone. The unit is 4 times of the PWM clock cycle.	0x0

6.13 UART

The Universal Asynchronous Receiver (UART) module is one of the serial I/O modules of the chip, and it is a full duplex Asynchronous system that can communicate with the peripheral device. Support for send and receive interrupts.

Base Address：0x40000000.

Name	地址	宽度	R/W	说明
RBR	0x00	32 Bits	R	Receive and send data registers

DLL	0x00	32 Bits	RW	Frequency division coefficient - low order
DLH	0x04	32 Bits	RW	Frequency division coefficient – high order
IER	0x04	32 Bits	RW	Interrupt Enable Register
IIR	0x08	32 Bits	R	Interrupt indicating register
LCR	0x0C	32 Bits	RW	control register
LSR	0x14	32 Bits	R	status register
USR	0x7C	32 Bits	R	UART status register

6.13.1 Receive and send data registers (RBR)

Name	RBR		
Offset	0x0		
Reset value	0x0000_0000		
Description			
Mapping	Type	Description	Reset value
[31:8]		reserved	0x0
[7:6]	Read only	LCR[7] bit = 0 is effective. For sending: When data is sent, the data to be sent is put into a register at this address. For receiving: After the data is received, the value of this address register is read as the received data. Note: send and receive correspond to two sets of registers.	0x0

6.13.2 Frequency division coefficient - low order (DLL)

Name	DLL		
Offset	0x0		
Reset value	0x0000_0000		
Description	Frequency division coefficient - low order		
Mapping	Type	Description	Reset value
[31:8]		reserved	0x0
[7:0]	W/R	LCR[7] bit = 1 is effective. Used to set baudrate. baud rate = (serial clock freq)/(16*divisor)	0x0

6.13.3 Frequency division coefficient - high order (DLH)

Name	DLH
Offset	0x4
Reset value	0x0000_0000

Description	Frequency division coefficient - high order		
Mapping	Type	Description	Reset value
[31:8]		reserved	0x0
[7:0]	W/R	LCR[7] bit = 1 is effective. Used to set baudrate. baud rate = (serial clock freq)/(16*divisor)	0x0

6.13.4 Interrupt Enable Register (IER)

Name	IER		
Offset	0x4		
Reset value	0x0000_0000		
Description	Interrupt Enable Register		
Mapping	Type	Description	Reset value
[31:2]		reserved	0x0
[1]	W/R	Enable the interrupt of sending.	0x0
[0]	W/R	Enable the interrupt of receiving.	0x0

6.13.5 Interrupt indicating register (IIR)

Name	IIR		
Offset	0x8		
Reset value	0x0000_0000		
Description	Interrupt indicating Register		
Mapping	Type	Description	Reset value
[31:4]		reserved	0x0
[3:0]	Read only	4’b0100: there’s data in the receive buffer. others： reserved	0x0

6.13.6 Control register (LCR)

Name	LCR		
Offset	0xC		
Reset value	0x0000_0000		
Description	Control register		
Mapping	Type	Description	Reset value
[31:4]		reserved	0x0
[7]	W/R	Divide factor access is valid	0x0
[6]	W/R	Failure control bit	0x0
[5]		reserved	0x0
[4]	W/R	Dipolar selection	0x0
[3]	W/R	polarity selection	0x0

[2]	W/R	Stop bit length 0:1 stop bit 1:1.5 stop bits	0x0
[1:0]	W/R	Data length settings: 00:5 bits 01:6 bits 10:7 bits 11:8 bits	0x0

6.13.7 Status register (LSR)

Name	LCR		
Offset	0x14		
Reset value	0x0000_0000		
Description	Status register		
Mapping	Type	Description	Reset value
[31:1]		reserved	0x0
[0]	Read only	data ready	0x0

6.13.8 UART status register(USR)

Name	USR		
Offset	0x7C		
Reset value	0x6		
Description	UART status register		
Mapping	Type	Description	Reset value
[31:1]		reserved	0x0
[0]	Read only	uart ready	0x0

6.14 Wakeup trigger Register

Name	WT_CTL		
Access Type	RW_REG		
Offset	0x00		
Reset value	0x0000_0000		
Description	Wakeup Trigger control		
Mapping	Domain	Description	Reset value
[31:20]		Reserved	0x0
[19:8]	deb_value	debounce time (32k clock) 0x0 : no debouncing Other: debouncing. The value will plus 1 by	0x0

		hardware. So, the Maximum is 0xFFE.	
[7:4]		Reserved	0x0
[3:1]		Reserved	0x0
[0]	wt_en	0:disable wakeup trigger 1:enable wakeup trigger	0x0

Name	WT_COMPARE		
Access Type	RW_REG		
Offset	0x04		
Reset value	0x0000_0000		
Description	Wakeup Timer compare register		
Mapping	Domain	Description	Reset value
[31:8]		Reserved	0x0
[7:0]	wt_compare	The number of events that have to be counted before the wakeup interrupt will be given	0x0

Name	WT_COUNTER		
Access Type	RO_REG		
Offset	0x08		
Reset value	0x0000_0000		
Description	Wakeup Timer event counter register		
Mapping	Domain	Description	Reset value
[31:8]		Reserved	0x0
[7:0]	wt_event_counter	This value represents the number of events that have been counted so far. It will be reset by resetting the interrupt	0x0

Name	WT_INPUT_EN		
Access Type	RW_REG		
Offset	0x0C		
Reset value	0x0000_0000		
Description	Wakeup Timer input enable register		
Mapping	Domain	Description	Reset value
[31:2]		Reserved	
[1:0]	wt_input_en	individual bit control	0x0

		0:input i_wt_in[x] is not enabled for wakeup event counter 1:input i_wt_in[x] is enabled for wakeup event counter	

Name	WT_POL		
Access Type	RW_REG		
Offset	0x10		
Reset value	0x0000_0000		
Description	Wakeup Timer input polarity control register		
Mapping	Domain	Description	Reset value
[31:2]		Reserved	
[1:0]	wt_pol	individual bit control 0: enable input i_wt_in[x] will increment the event counter if that input goes low 1: enable input i_wt_in[x] will increment the event counter if that input goes high	0x0

Name	WT_INT_STS		
Access Type	INT_STS		
Offset	0x20		
Reset value	0x0000_0000		
Description	wakeup timer interrupt status		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]	wt_int_STS	wakeup timer interrupt	0x0

Name	WT_INT_EN		
Access Type	INT_EN		
Offset	0x24		
Reset value	0x0000_0000		
Description	wakeup timer interrupt enable		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]	wt_int_EN	wakeup timer interrupt	0x0

Name	WT_INT_MSK		
Access Type	INT_MSK		
Offset	0x28		
Reset value	0x0000_0000		
Description	wakeup timer interrupt mask		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]	wt_int_MSK	wakeup timer interrupt	0x0

Name	WT_INT_CLR		
Access Type	INT_CLR		
Offset	0x2C		
Reset value	0x0000_0000		
Description	wakeup timer interrupt clear		
Mapping	Domain	Description	Reset value
[31:1]		reserved	0x0
[0]	wt_int_CLR	wakeup timer interrupt	0x0

6.15 WDT Register

6.15.1 Load-value register – High-8-bits

Name	watchdog_comp_value_h		
Access Type	RW_REG		
Offset	0x64		
Reset value	0xFF		
Description	watchdog comparision high byte value		
Mapping	Domain	Description	Reset value
[31:8]		reserverd	0x0
[7:0]	wdg_comp_val_h	WDT: High 8 bits of the load value.	0xff

6.15.2 Load-value register – Mid-8-bits

Name	watchdog_comp_value_m
Access Type	RW_REG
Offset	0x68
Reset value	0xFF
Descriptio	watchdog comparision middle byte value

n			
Mapping	Domain	Description	Reset value
[31:8]		reserverd	0x0
[7:0]	wdg_comp_val_m	WDT: Mid 8 bits of the load value.	0xff

6.15.3 Load-value register – Low-8-bits

Name	watchdog_comp_value_l		
Access Type	RW_REG		
Offset	0x6c		
Reset value	0xFF		
Description	watchdog comparision low byte value		
Mapping	Domain	Description	Reset value
[31:8]		reserverd	0x0
[7:0]	wdg_comp_val_l	WDT: Low 8 bits of the load value.	0xff

6.15.4 Load-value latch register

Name	watchdog_comp_vld		
Access Type	WC		
Offset	0x70		
Reset value	0x00		
Description	watchdog comparision value valid		
Mapping	Domain	Description	Reset value
[31:1]		reserverd	0x0
[0]	wdg_comp_vld	1:watchdog latch the load value	0x0

6.15.5 Counter-clear Register

Name	watchdog_feed_dog		
Access Type	WC		
Offset	0x74		
Reset value	0x00		
Description	watchdog feed dog		
Mapping	Domain	Description	Reset value
[31:1]		reserverd	0x0
[0]	wdg_feed	watchdog feed dog	0x0

6.15.6 Interrupt-clear Register

Name	watchdog_interrupt_clear		
Access Type	WC		
Offset	0x78		
Reset value	0x00		
Description	watchdog interrupt clear		
Mapping	Domain	Description	Reset value
[31:1]		reserverd	0x0
[0]	wdg_int_clr	watchdog interrupt clear	0x0

6.15.7 Counter-High Register

Name	watchdog_counter_value_h		
Access Type	RO_REG		
Offset	0x7c		
Reset value	0x00		
Description	watchdog counter value high byte		
Mapping	Domain	Description	Reset value
[31:8]		reserverd	0x0
[7:0]	wdg_counter_val_h	watchdog High-8-bits of the Counter	0x0

6.15.8 Counter-Mid Register

Name	watchdog_counter_value_m		
Access Type	RO_REG		
Offset	0x80		
Reset value	0x00		
Description	watchdog counter value middle byte		
Mapping	Domain	Description	Reset value
[31:8]		reserverd	0x0
[7:0]	wdg_counter_val_m	watchdog Mid-8-bits of the Counter	0x0

6.15.9 Counter-Low Register

Name	watchdog_counter_value_l		
Access Type	RO_REG		
Offset	0x84		
Reset value	0x00		
Description	watchdog counter value low byte		
Mapping	Domain	Description	Reset value
[31:8]		reserverd	0x0
[7:0]	wdg_counter_val_l	watchdog counter value low byte	0x0

6.15.10 Status Register

Name	watchdog_state		
Access Type	RO_REG		
Offset	0x88		
Reset value	0x04		
Description	watchdog counter value low byte		
Mapping	Domain	Description	Reset value
[31:5]		reserverd	0x0
[4]	wdg_flag	watchdog reset flag	0x0
[3]	wdg_int	watchdog int	0x0
[2]	wdg_int_en	watchdog int enable,low is assert	0x1
[1:0]	wdg_fsm_val	watchdog state mechine value	0x0

6.16 LVD Register

6.16.1 Control Register

Name	module_mode		
Access Type	RW_REG		

Offset	0x8C		
Reset value	0x18		
Description	rf bb mode		
Mapping	Domain	Description	Reset value
[31:6]		reserverd	0x0
[5]	en_lvd_ull	enable lvd detect, high valid	0x0
[4:3]	cpu_clk_sel	cpu clock sel. 0:2M, 1:4M, 2:8M ,3: 16M	0x3
[2]	watch_dog_en	watch dog enable, high valid	0x0
[1:0]	rf_bb_mode	00:chip rf and chip bb mode. 01:external rf and chip bb mode. 10:chip rf and external bb mode.	0x0

6.16.2 Module Register

Offset	0x4012018C		
Mapping	Domain	Description	Reset value
[5]	en_lvd_ull	Enable LVD，High is effective.	0x0

Offset	0x40120100		
Mapping	Domain	Description	Reset value
[1]	lvd_pwr_ctrl_en	Enable LVD to wake up from sleep mode, high is effective.	0x0

6.17 Radio data Register

The offset address of the baseband register is 0x4007_0000. RX after the register name in the following table indicates that the register is used for receiving, and TX indicates that it is used for transmitting. The access types WC/RW/RO respectively indicate write pulse signal, read and write, and read only. The default configuration values of the baseband registers are given in the table, and the user can use software configuration to make changes according to actual needs.

6.17.1 bb_soft_rst

Name	bb_soft_rst (TX/RX)
Access Type	WC

Offset	0x00		
Reset value	0x01		
Mapping	Domain	Description	Reset value
[31:1]		reserverd	0x0
[0]	bb_soft_rst	BB soft reset, high is effective.	0x1

Note: It is not necessary to modify the register for soft-reset after the data packet is sent and received. Soft reset is necessary to modify the register during the sending and receiving process to prevent the baseband state machine from being in the previous register configuration mode and causing data errors.

6.17.2 ac_code_mode

Name	ac_code_sel (RX)		
Access Type	RW		
Offset	0x3C		
Reset value	0x00		
Mapping	Domain	Description	Reset value
[31:1]		reserverd	0x0
[0]	dual_syncword_mode	1: syncword1 and syncword2 match together; 0: syncword1 match only.	0x0

Note: bb supports up to two addresses at the same time to do correlation matching. At this time, the register is set to 1, the one that is related first will close the other one, and it is considered to be syncword1 related when it is related.

When this register is set to 0, only syncword1 is used for correlation matching, at this time syncword2 is invalid.

6.17.3 ac_code1

Name	ac_code1 (TX/RX)		
Access Type	RW		
Offset	0x40		
Reset value	0xAAAA_AAAA		
Mapping	Domain	Description	Reset value
[31:0]	cs_syncword1_lowpart	syncword1, low32 bits	0xAAAA_AAAA

Note: In transmit mode, this register represents the sending address, and it is filled from the low bit, that is, when the address is 3byte, only the low 24bits of this register are used.

In receiving mode, this register represents the lower 32bits of syncword1. The receiving mode does not distinguish between address and preamble. The lower 32bits of the two are filled in this register.

6.17.4 ac_code2

Name	ac_code2 (RX)		
Access Type	RW		
Offset	0x44		
Reset value	0xAAAA_AAAA		
Mapping	Domain	Description	Reset value
[31:0]	cs_syncword2_lowpart	syncword2, low 32 bits	0xAAAA_AAAA

Note: The configuration value of this register is valid only when the two addresses of the receiving mode are used for correlation matching (ie register dual_syncword_mode =1).

6.17.5 ac_code_highpart

Name	ac_code_highpart (RX)		
Access Type	RW		
Offset	0x48		
Reset value	0xAAAA		
Description	access code1(syncword1) lowpart		
Mapping	Domain	Description	Reset value
[31:16]		reserverd	0x0
[15:8]	cs_syncword2_highpart	syncword2, high 8 bits	0xAA
[7:0]	cs_syncword1_highpart	syncword1, high 8 bits	0xAA

Note: In the receiving mode, this register represents the high 8bits of syncword1 and syncword2 (if dual_syncword_mode =1). The receiving mode does not distinguish between address and preamble. The high 8bits of the two are filled in this register.

6.17.6 bit_stream

Name	bit_stream (TX/RX)		
Access Type	RW		
Offset	0x5C		
Reset value	0x255_5555		
Mapping	Domain	Description	Reset value
[31:26]		reserverd	0x0
[25]	whit_dsb	0: whitening; 1: disable whitening	0x1
[24]	crc_dsb	0: enable CRC; 1: disable CRC	0x0
[23:0]	cs_crcinit	CRC24 initial value	0x55_5555

6.17.7 tx_type_len

Name	tx_type_len (TX)
Access Type	RW

Offset	0x60		
Reset value	0x2025		
Mapping	Domain	Description	Reset value
[31:14]		reserverd	0x0
[13]	tx_data_invert	Inverse bits of the send-data	0x1
[12:9]		reserverd	0x0
[8]	txrxadd		0x0
[7]	txtxadd		0x0
[6:0]	txadvlen	Gets the length of the sent packet from the CPU	0x25

Note: tx_data_invert is the inversion of the sending bit and is fixed to 1.

txrxadd and txtxadd indicate the address type in standard Bluetooth, public or random, and the definition refers to the Bluetooth standard protocol. It is only used in the test mode, and it does not care about the configuration under normal transmission.

Txadvlen is the set sending data length. In the test mode, this value, txrxadd and txtxadd will be spliced into the Length part of the packet format Header. **Under normal working conditions, txadvlen is set to the length +2 in the Header in the standard specification.** This is because the Header and Payload in the specification are no longer distinguished under normal working conditions. +2 is the number of bytes occupied by the Header in the specification. The length defined by the Header in the standard specification is only the length of the payload, which is 2 different from the txadvlen here.

Taking the default value of 0x25 in the table as an example, under normal working conditions, after removing the 2 bytes of the Header, only 35 payloads instead of 37 bytes can be transmitted. When Header is also used as Payload, only 37 are counted.

6.17.8 rx_tx_sel

Name	rx_tx_sel (TX/RX)		
Access Type	RW		
Offset	0x64		
Reset value	0x00		
Mapping	Domain	Description	Reset value
[31:1]		reserverd	0x0
[0]	pktcntl_rxtxn	0: TX; 1: RX	0x0

6.17.9 data_channel_idx

Name	data_channel_idx (TX/RX)
Access Type	RW
Offset	0x68
Reset value	0x25

Mapping	Domain	Description	Reset value
[31:6]		reserverd	0x0
[5:0]	data_ch_idx	Channel number	0x25

Note: The channel number here is data_channel_idx, which is different from rf_channel_idx, which is arranged in sequence from 0 to 39. The broadcast addresses 37, 38, and 39 of data_chanel correspond to 0, 12, and 39 of rf_channel, and the rest of the data_channel interval is distributed in 1~11, 13~38 of rf_channel and arranged in sequence.

6.17.10 start_p

Name	start_p (TX/RX)		
Access Type	WC		
Offset	0x70		
Reset value	0x00		
Mapping	Domain	Description	Reset value
[31:1]		reserverd	0x0
[0]	pktcntl_start_p	BB starts pulse mark.	0x0

Note: **Only single buffer mode and streaming mode need to use this register, double buffer mode (FIFO mode) does not need.**

6.17.11 tx_mode_sel

Name	tx_mode_sel (TX)		
Access Type	RW		
Offset	0x74		
Reset value	0x1AA_AAAA		
Mapping	Domain	Description	Reset value
[31:28]		reserverd	0x0
[27:26]	cs_txpreamble_mode	Send leading code mode 0:1 bytes; 1: 2 bytes; 2:3 bytes	0x0
[25:24]	cs_txsyncword_mode	Send syncword-mode 0:3 bytes; 1: 4 bytes; 2:5 bytes	0x1
[23:0]	cs_preamble	Send premble	0xAA_AAAA

Note: Only used in transmit mode, syncword = preamble + address. The selection of cs_preamble selects the corresponding bit according to cs_txpreamble_mode. For example, when cs_txpreamble_mode = 1, the lower 8 bits of cs_preamble are valid.

The address sent is in the R40 register.

6.17.12 bb_state_cntl5

Name	bb_state_cntl5 (TX/RX)

Access Type	RW		
Offset	0x84		
Reset value	0x40_0000		
Mapping	Domain	Description	Reset value
[31]	soft_en_trx	At launch, the software is configured to enable BB	0x0
[30]	soft_rxb_tx	At launch, the software configures receiving/transmitting options	0x0
[29]	soft_en_pa	During launch, the software is configured to enable launch data	0x0
[28]	soft_start_rx_adc	At launch, the software configuration starts the ADC	0x0
[27:20]	txpwrdn	Transmit power-on delay	0x4
[19:16]	tx_test_mode_type	Emission data TYPE in test mode： 0x0 : PRBS9; 0x1 : EYE LONG1. i.e MSB to LSB “00001111”; 0x2 : EYE SHORT1, i.e MSB to LSB “01010101”; 0x3 : PRBS15; 0x4 : ALL 1; 0x5 : ALL 0; 0x6 : EYE LONG0, i.e MSB to LSB “11110000”; others : EYE SHORT0, i.e MSB to LSB “10101010”;	0x0
[15:9]	rxlen_cpu1	Syncword2 corresponds to the length of the received data	0x0
[8:2]	rxlen_cpu0	Syncword2 corresponds to the length of the received data	0x0
[1:0]	rx_length_mode	Receiving data length value mode ➢ 0: Obtained from the received packet demodulation; ➢ 1: Obtained from the register configuration	0x0

Note: bit[31:16] is used for testing, users do not need to change it during normal use.

When rx_length_mode=0, analyze the data packet format according to the received data packet to

obtain the received data length.

When rx_length_mode =1, the received data length is directly set to rxlen_cpu0 or rxlen_cpu1.

6.17.13 reg2ctrl_buf

Name	reg2ctrl_buf (TX/RX)		
Access Type	WC		
Offset	0x94		
Reset value	0x00		
Mapping	Domain	Description	Reset value
[31:9]		reserverd	0x0
[1]	ctrl2reg_buf_rfin	Buffer read off	0x0
[0]	ctrl2reg_buf_wfin	Buffer write off	0x0

Note: In the transmit mode, pull up wr_finish, indicating that the data in the Buffer is ready, and the baseband can read the data in the Buffer (for sending operations), otherwise the baseband thinks that the data is not ready.

In the receiving mode, raise rd_finish, indicating that the data in the buffer has been processed, and the baseband can write new data into the data buffer.

6.17.14 reg_interrupt

Name	reg_interrupt (TX/RX)		
Access Type	RW		
Offset	0x98		
Reset value	0x555		
Mapping	Domain	Description	Reset value
[31:12]		reserverd	0x0
[11]	syncerr_int_mask	syncerr interrupt mask	0x0
[10]	syncerr_int_en	syncerr interrupt enable	0x1
[9]	typeerr_int_mask	typeerr interrupt mask	0x0
[8]	typeerr_int_en	typeerr interrupt enable	0x1
[7]	lenerr_int_mask	lenerr interrupt mask	0x0
[6]	lenerr_int_en	lenerr interrupt enable	0x1
[5]	crcerr_int_mask	crcerr interrupt mask	0x0
[4]	crcerr_int_en	crcerr interrupt enable	0x1
[3]	reg2ctrl_rx_int_mask	rx interrupt mask	0x0
[2]	reg2ctrl_rx_int_en	rx interrupt enable	0x1
[1]	reg2ctrl_tx_int_mask	tx interrupt mask	0x0
[0]	reg2ctrl_tx_int_en	tx interrupt enable	0x1

Note: Only when both mask and enable are 1, the corresponding interrupt is valid.

6.17.15 reg_interrupt_RO

Name	reg_interrupt_RO (TX/RX)		
Access Type	RO		
Offset	0x9C		
Reset value	0x00		
Mapping	Domain	Description	Reset value
[31:28]		reserverd	0x0
[27:20]	buf1_rssi	output rssi	0x0
[19]	buf1_acsync_found_addr_num	syncword no1 or no2	0x0
[18:11]	buf0_rssi	output rssi	0x0
[10]	buf0_acsync_found_addr_num	syncword no1 or no2	0x0
[9]	buf1_syncerr_raw_int_state	syncerr raw interrupt state	0x0
[8]	buf1_typeerr_raw_int_state	typeerr raw interrupt state	0x0
[7]	buf1_lenerr_raw_int_state	lenerr raw interrupt state	0x0
[6]	buf1_crcerr_raw_int_state	crcerr raw interrupt state	0x0
[5]	buf0_syncerr_raw_int_state	syncerr raw interrupt state	0x0
[4]	buf0_typeerr_raw_int_state	typeerr raw interrupt state	0x0
[3]	buf0_lenerr_raw_int_state	lenerr raw interrupt state	0x0
[2]	buf0_crcerr_raw_int_state	crcerr raw interrupt state	0x0
[1]	ctrl2reg_rx_raw_int_state	rx raw interrupt state	0x0
[0]	ctrl2reg_tx_raw_int_state	tx raw interrupt state	0x1

Note: There are two buffers in the baseband, both of which can store interrupt information. acsync_found_addr_num indicates the address of the relevant match, 0: the first address is matched, 1: the second address is matched.

6.17.16 bbram_mode

Name	bbram_mode (TX/RX)		
Access Type	RW		
Offset	0xA0		
Reset value	0x02		
Mapping	Domain	Description	Reset value
[31:3]		reserverd	0x0
[2:0]	bbram_mode	bbram control mode 0: double-buf, bb-write, cpu-read; 1: double-buf, bb-read, cpu-write; 2: single-buf, bb-write, cpu-read;	0x2

		3: single-buf, bb-read, cpu - write; 4：double-buf send/receive.	

Note: single buffer transmission, bbram_mode = 3; continuous transmission, bbram_mode = 1;
Single buffer reception, bbram_mode = 2; continuous reception, bbram_mode = 0;
Dual-buffer transceiver, bbram_mode = 4, requires software to control the transceiver to switch continuously.

6.17.17 buf_stats

Name	buf_stats (TX/RX)		
Access Type	RO		
Offset	0xA4		
Reset value	0x00		
Mapping	Domain	Description	Reset value
[31:9]		reserverd	0x0
[8:2]	ctrl2reg_buf_num	Buffer serial number	0x0
[1]	ctrl2reg_buf_empt	Buffer null indicator	0x0
[0]	ctrl2reg_buf_full	Buffer full indicator	0x0

Note: When the buffer is empty, it cannot be read and no more data can be received.
When the buffer is full, it cannot be written and no more data can be transmitted.
Buffer_num indicates the number of two pieces of bbram, 0 is the first piece of ram, and 64 is the second piece of ram.

6.17.18 intp_cls

Name	intp_cls (TX/RX)		
Access Type	WC		
Offset	0xAC		
Reset value	0x00		
Mapping	Domain	Description	Reset value
[31:10]		reserverd	0x0
[9]	buf1_syncerr_int_cls	buf1: Receive related error interrupt cleared	0x0
[8]	buf1_typeerr_int_cls	buf1: Receive Type error interrupt cleared	0x0
[7]	buf1_lenerr_int_cls	buf1: Receive length error interrupt cleared	0x0
[6]	buf1_crcerr_int_cls	buf1: Receive CRC check interrupt cleared	0x0
[5]	buf0_syncerr_int_cls	buf0: Receive related error interrupt cleared	0x0
[4]	buf0_typeerr_int_cls	buf0: Receive Type error	0x0

		interrupt cleared	
[3]	buf0_lenerr_int_cls	buf0: Receive length error interrupt cleared	0x0
[2]	buf0_crcerr_int_cls	buf0: Receive CRC check interrupt cleared	0x0
[1]	reg2ctrl_rx_int_cls	Receive interrupt cleanup	0x0
[0]	reg2ctrl_tx_int_cls	Transmit interrupt cleanup	0x0

Note: After the interrupt of the corresponding data in buf0 and buf1 is cleared, the next interrupt can be triggered by the corresponding buffer.

After the transmit interrupt and receive interrupt are cleared, the next interrupt can be triggered.

7 RAM

Type	Start-Addr	Size
RAM(Program Run)	0x20000000	2K Bytes
RAM(Radio data)	0x50000000	128 Bytes
RAM(Retation)	0x60000000	16 Bytes

Table 21. RAM address

8 Working/Sleep Mode

The AK801 has 2 working states: Working mode and sleep-down mode. Modes can be switched by registers or external triggers. After AK801 starts up, it enters the working mode. In the working mode, MCU enters the sleep mode through register configuration. In the sleep mode, MCU can enter working mode through TIMER\IO-Input\VDD-Detection.

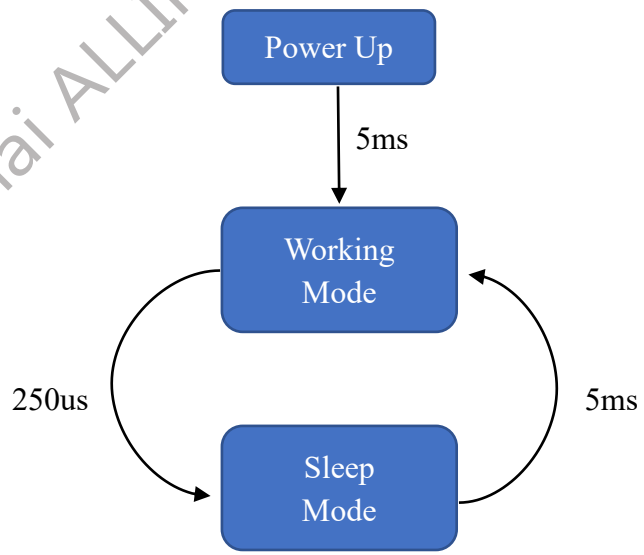


Figure 22. Mode Switch

8.1 Sleep Mode

Sleep mode makes the AK801 to consume the lowest power, the internal 32K clock and a few

circuits are still working. Retention-Memory(16 Bytes) can hold some states in sleep mode. AK801 also retains the IO Function-mode, pull-up state, direction state.

In the sleep mode, WDT/IO-Detection circuit/VDD-Detection circuit are all in the state of working mode. You can configure these modules to wake up the chip and make the AK801 enter the working mode. The configuration of WDT can wake up the chip after a period of time, and the configuration of IO level detection circuit can detect the change of IO input level state, so as to wake up the chip. VDD voltage detection found that the VDD voltage exceeds a certain voltage threshold can wake up the chip.

For the specific configuration of the three wake-up modes, you can refer to the corresponding module description respectively.

	MCU	RF	Baseband	Watch dog	Wake up timer	LVD	Others
Working mode	√	√	√	√	√	√	√
Sleep mode	*	*	*	√	√	√	*

Bit [0] of register POWER_CTRL_REG is set to 1 to enter sleep mode. In sleep mode, in order to minimize current consumption, all I/O pins should remain VDD or GND, with no external circuits consuming current from I/O pins.

To avoid switching current caused by the input pin being suspended, the high resistance input I/O pin should be externally pulled to high or low level. In order to minimize current consumption, the influence of the pull-up resistance inside the chip should also be considered.

8.2 Working Mode

In the working mode of AK801, all modules of the chip are in the power state. There are two ways to enter the working mode of AK801. One is to directly enter the working mode after the chip is powered on, and the other is to enter the working mode from the sleep mode.

The chip provides a set of registers for the MCU to query whether it has entered the working mode from sleep mode, and if it has been awakened from sleep mode, it can query which way to end the sleep state.

9 CPU

Reference: ***CK 801 User Manual***.

10 Interrupt

10.1 Overview

The interrupt controller has 12 external interrupt sources, each with a separate software-programmable interrupt priority. The vector interrupt controller collects the interrupt requests from

different interrupt sources and arbitrates the interrupt requests according to the interrupt priority. The highest priority interrupt takes control of the interrupt and makes an interrupt request to the processor.

Interrupt controller supports interrupt nesting. When a higher priority interrupt request comes in while the processor is processing an interrupt request, the processor will respond to the higher priority interrupt request by interrupting the processing of the current interrupt service routine. At the end of higher priority interrupt request processing, the CPU returns the interrupted interrupt service routine to continue execution. Vector interrupt controller allows interrupt requests of high priority to preempt interrupt requests of low priority, but does not allow interrupt requests of the same level or low priority to preempt, so the real-time response of interrupt is guaranteed.

10.2 Interrupt mechanism

Vector interrupt controller supports level interrupt and pulse interrupt.

For level interrupts, the vector interrupt controller sets the corresponding interrupt into the waiting state after sampling the high level of the valid interrupt signal, and then requests the CPU to respond. The level interrupt requires the interrupt service routine to clear the valid signal of the interrupt source of the peripheral, otherwise the interrupt controller will re-issue the interrupt request to the CPU when the interrupt exits. A peripheral can take advantage of this by leaving the interrupt signal constant until it no longer needs to be processed by the interrupt handler.

For pulse interrupts, also known as edge interrupts, the vector interrupt controller samples the rising edge of the effective interrupt signal, and then sets the corresponding interrupt into the waiting state, and then initiates the interrupt request to the CPU. To ensure that the vector interrupt controller detects a pulse interrupt, the periphery needs to hold the interrupt signal for at least one CPU clock cycle. If the pulse interrupt source sends multiple interrupt requests to the vector interrupt controller before the CPU responds to the impulse interrupt request, the vector interrupt controller will only record one interrupt request. After the CPU responds to the pulse interrupt request, if the pulse interrupt source makes another request to the interrupt controller, the vector interrupt controller will trigger the corresponding interrupt to enter the waiting state again, and the interrupt request in the waiting state can only be responded by the CPU again after the last interrupt exits.

In addition, the vector interrupt controller supports software interrupts. By setting the interrupt setting wait register (VIC_ISPR) to raise the corresponding interrupt waiting state bit, the software can trigger the interrupt to enter the waiting state and send the interrupt request to the CPU.

When the processor responds to the interrupt request, the vector interrupt controller automatically clears the corresponding interrupt's wait state bits. It is also possible to clear the wait status bits of the corresponding interrupt by setting the interrupt clear wait register (VIC_ICPR). For level interrupts, if the interrupt valid signal continues to be high, the wait state bits cannot be cleared by setting the VIC_ICPR register.

10.3 Interrupt status bits

VIC provides 2 status bits for each interrupt source, as follows:

- Pending: the interrupt is in a waiting state, namely the interrupt request is waiting for the CPU to respond.
  - 0: The interrupt is not yet in a wait state;
  - 1: The interrupt has been in a wait state.
- Active: the interrupt request has been responded by the CPU, but has not been processed.
  - 0: The interrupt request has not been responded by the CPU;
  - 1: The interrupt request has been responded by the CPU, but has not yet been processed.

Conditions for setting the Pending bit:

- Level interrupt source, interrupt source effective signal is set high, and ACTIVE is low; or CPU is exiting the interrupt service routine;
- Pulse interrupt source, rising edge is effective;
- Set ISPR.

The Pending bit is cleared by:

- The CPU responds to the interrupt request;
- Software clears ICPR.

Conditions for setting Active bits:

- The CPU responds to the interrupt request.

ACTIVE bit clearance condition:

- The CPU exits the interrupt service routine.

Note: For level interrupt source, since it is necessary for the interrupt service routine to pull down the level interrupt source request, it is only necessary to sample the valid signal of the interrupt source and set the Pending bit when ACTIVE is low or the interrupt service routine is exiting. For the pulse interrupt source, the rising edge of the pulse interrupt signal should be sampled in real time and the Pending bit should be set because the request signal will be automatically pulled down.

10.4 Interrupt priority

VIC provides four priority settings for each interrupt source through the interrupt priority setting register (IPR0~IPR7). The lower the priority number, the higher the priority, refer to Section 10.2 for details. When the software does not set the priority register, all interrupt sources default to the highest priority -- 0.

When multiple Pending interrupts have the same priority number, the order of interrupt submission will be determined according to the interrupt number, and the smaller interrupt number will be submitted to the CPU first. For example, two interrupt requests, IRQ0 and IRQ1, have the same priority number. The interrupt source number of IRQ0 is smaller than that of IRQ1, so IRQ0 is submitted to the CPU first.

10.5 Interrupt vector number

The interrupt vector number is the position number of the interrupt request in the anomaly vector scale. The following figure shows the abnormal vector scale of CK801.

The initial 0 to 30 vectors are used for processor internal recognition; The 31 vector is reserved; The vector numbers starting from 32 are external interrupt requests, and each interrupt source corresponds to an interrupt vector number.

Vector	Offset	Description
0	00	Restart Exception
1	04	Unaligned Exception
2	08	Access Error Exception
3	0C	Reserved
4	10	Illegal Instruction Exception
5	14	Privilege Violation Exception
6	18	Reserved
7	1C	Breakpoint Exception
8	20	Unrecoverable Error Exception
9~15	24~3C	Reserved
16~19	40~4C	Trap Instruction Exception (Trap # 0-3)
20~30	50~78	Reserved
31	7C	Reserved
32	0x80	Watchdog Interrupt
33	0x84	Timer0 Overflow Interrupt
34	0x88	Timer1 Overflow Interrupt
35	0x8C	Timer2 Overflow Interrupt
36	0x90	UART Overflow Interrupt
37	0x94	GPIO Interrupt
38	0x98	Radio Receive Interrupt
39	0x9C	Radio Transmit Interrupt
40	0xA0	Catch Comparator Interrupts
41	0xA4	Wake-up Interrupt
42	0xA8	LVD Interrupt
43	0xAC	Core timer Overflow Interrupt

10.6 Interrupt processing

Interrupt processing process can be divided into the following steps:

- Interrupt request: Interruption in a waiting state sends an interrupt request to CPU after the priority arbitration.

- Interrupt response: CPU responded with an interrupt on instruction exit. Update the PSR and PC to the EPSR and EPC; Update PSR.VEC with the interrupt vector number that was responded to the interrupt; Clear the PSR.EE; Get the entry address of exception; According to the interrupt response signal, the Pending state bit of the corresponding interrupt is cleared, and its Active state bit is set.
- Interrupt state save: Save the interrupt control register field {EPSR, EPC} firstly; Open the PRS.EE and PRS.IE, enable the Interrupt Nest; Then save the universal register states.
- Interrupt Transaction: The CPU starts processing the interrupt transaction. For level interrupts, it is necessary to clear the interrupt source signal, otherwise the interrupt will be reentered when the interrupt exits;
- Recovery the interrupt states and Exit: First restore the general-purpose registers states; Then restore the interrupt control register field {EPSR, EPC}, restore the EPC and EPSR to PC and PSR, and exit the interrupt service;

The interrupt state can be saved by executing the NIE and IPUSH instructions at the beginning of the interrupt routine, and the recovery and exit of the interrupt site can be performed by executing the IPOP and NIR instructions at the end of the interrupt service routine.

10.7 Interrupt Nest

AK80X supports interrupt nesting, allowing higher priority interrupt preemption in the process of interrupt processing, thus improving the real-time response of interrupt. When multi-level interrupt nesting exists, it may be necessary to change the priority of preempted interrupts in certain scenarios, such as low priority interrupt response time reaching maximum limit. At this time, the software can improve the priority condition of interrupt preemption by setting the interrupt priority threshold register, so that the preempted low-priority interrupts can get timely response.

10.7.1 Interrupt nesting priority condition

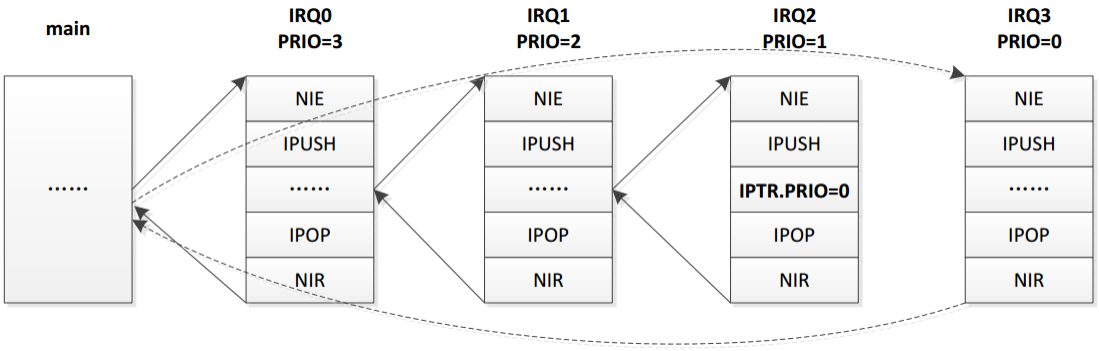
The priority conditions for interrupt preemption can be divided into two categories, as follows:

- when the interrupt priority threshold is not enabled, the interrupt preemptive priority must be higher than which the current CPU is processing; Peer priority cannot be preempted;
- When the interrupt priority threshold is enabled, the priority of interrupt preemption must not only be higher than the interrupt priority currently being processed by the CPU, but also higher than the threshold set by the interrupt priority threshold register.

AK80X supports dynamic adjustment of interrupt priority. When the nested interrupt priority needs to be moved up or down, the interrupt priority configuration register is set at the same time as the interrupt priority threshold register.

The following figure shows an example of interrupted preemption. The interrupt priority is set as: IRQ0<IRQ1<IRQ2<IRQ3; The order in which interrupt source requests are generated is: IRQ0>IRQ1>IRQ2>IRQ3. The CPU responds to IRQ0 first, and during the execution of IRQ0 interrupt service routine, higher priority IRQ1 arrives, so IRQ0 is preempted, and the CPU starts to

execute IRQ1 interrupt service routine. Similarly, IRQ2 preempts IRQ1 and sets the interrupt priority threshold register (IPTR.VECTTHRESHOLD=IRQ0, IPTR.PRIOTHRESHOLD=0 , IPTR.EN=1). When IRQ3 arrives, although the priority is higher than IRQ2, the priority of IRQ3 is not higher than the interrupt represented by the priority threshold set in IPTR, so IRQ3 cannot preempt IRQ2. IRQ3 does not get a CPU response until IRQ0's interrupt service routine has finished executing and cleared IPTR.EN.



10.7.2 Timing conditions for interrupt nesting

Under the condition of satisfying the priority of interrupt preemption, it is also necessary to determine the phase of the current interrupt response. The process of interrupt response is shown in 3.3.4, which is mainly divided into the following stages related to nesting: 1) update of EPSR, EPC, PSR and read of abnormal entry address, 2) NIE instruction, 3) IPUSH instruction, 4) interrupt transaction, 5) IPOP instruction, and 6) NIR instruction.

To ensure that the interrupt nested scene is saved and restored, the CPU cannot be interrupted by interrupts in the following stages:

- Update the EPSR, EPC, PSR, and get the exception entry address during the process after interrupt response;
- NIE instruction execution process, including instruction retirement;
- PSR.IC bit closed, IPUSH and IPOP in the course of instruction execution, not including instruction retirement;
- NIR instruction execution process does not include instruction retirement.

The CPU can safely and reliably respond to new interrupts in the following phases:

- In the course of normal program execution, before the interrupt response;
- IPUSH, IPOP retired;
- PSR.IC bit is open, IPUSH, IPOP instructions are in the process of execution;
- NIR retired;
- During the processing of an interrupt transaction.

When the PSR.IC bit is turned on, the IPUSH and IPOP instructions can respond to interrupts during execution, so the instructions need to be re-executed when the interrupts return. If the IPUSH or IPOP responds to interruption during retirement, the next instruction of IPUSH /IPOP

will be directly executed after the interruption exits, and there is no need to re-execute the IPUSH /IPOP command. NIR instruction can't be interrupted by interrupts during execution, but can respond to interrupts at retirement. If the interrupt is hit on the NIR instruction, the CPU will directly push the return address of NIR onto the stack when the NIR instruction retires, and return the target address of NIR when the interrupt exits.

11 GPIO

AK801 series chips have up to 20 GPIOs. The number of GPIOs varies with the chip model. GPIO supports several functions, which can be selected through the PAD_MUX register.

Most AK801 pins support multiple functions, which can be used as either general IO or special functions, and the functions can be selected by configuring the corresponding registers.

Pin-reuse is shown in the following table, where "programming function" is determined by on-chip mode, and other functions can be configured by registers.

IO	Function 0		Function 1		Function 3		Function 6		Program Function	
	Function	Direction	Function	Direction	Function	Direction	Function	Direction	Function	Direction
GPIO00	GPIO	B	urt_rxd	I		O	Wakeup IO	I		
GPIO01	GPIO	B	urt_txd	O		O	Wakeup IO	I		
GPIO02	GPIO	B				O	Wakeup IO	I	prog_seri_data_0	B
GPIO03	GPIO	B	pwm0	O		O	Wakeup IO	I	prog_seri_data_1	B
GPIO04	GPIO	B	pwm1	O	jtag_tms	B	Wakeup IO	I	prog_seri_data_en	I
GPIO05	GPIO	B	pwm2	O	jtag_clk	I	Wakeup IO	I	prog_mode_en	I
GPIO06	GPIO	B	pwm3	O	jtag_trst_b	I	Wakeup IO	I	prog_pweb	I
GPIO07	GPIO	B	pwm4	O	tccr_ex	I	Wakeup IO	I	otp_clk	I
GPIO08	GPIO	B	pwm5	O			Wakeup IO	I		
GPIO09	GPIO	B	jtag_tms	B			Wakeup IO	I		
GPIO10	GPIO	B	jtag_clk	I			Wakeup IO	I		
GPIO11	GPIO	B	jtag_trst_b	I			Wakeup IO	I		
GPIO12	GPIO	B					Wakeup IO	I		
GPIO13	GPIO	B					Wakeup IO	I		
GPIO14	GPIO	B			urt_rxd	I	Wakeup IO	I		
GPIO15	GPIO	B			urt_txd	O	Wakeup IO	I		
GPIO16	GPIO	B					Wakeup IO	I		
GPIO17	GPIO	B					Wakeup IO	I		
GPIO18	GPIO	B					Wakeup IO	I		
GPIO19	GPIO	B					Wakeup IO	I		

11.1 General I/O (GPIO)

GPIO is a 20-bits wide bidirectional port that can be set to both input and output modes. When

configured as output, the value written to the data register is printed to the corresponding I/O pin; The input data register captures the data on the I/O pins each clock cycle. GPIO-Pins have a configurable internal pull-up function. GPIO 0/1 must also be configured with the top level control register (TOP_CTRL_REG).

Register	TOP_CTRL_REG		
Bit	Domain	Description	Reset Value
[6:5]	pin0_mode_ull	GPIO Function 00: GPADC Input 01: Reserved 10: Digital Input 11: Digital Output	0x2
[4:3]	pin1_mode_ull	GPIO Function 00: GPADC Input 01: Reserved 10: Digital Input 11: Output	0x2

Note:

- The GPIO02 is lowered externally when the chip is powered on, which can put the chip into debug/programming mode, where the MCU does not execute any instructions.
- GPIO 00/01/19 has no pull-up function.
- GPIO19 is an open drain pin. When the pin is output, it can only output low level. If it needs to output high level, it needs to connect an external pull resistor.

11.2 Function reuse

Different functions can be configured through the multiplexing function register (PAD_GPIO_MUX), and internal weak pull-ups can be activated and disabled. The default is set to Function_0, which is the general I/O function.

If the port is set to multiplexing, the pin and the output register are disconnected and connected to the output signal of the on-chip peripheral. If the GPIO pin is configured to multiplexing output but the peripheral is not activated, its output will be uncertain.

11.3 GPIO interrupt

All GPIO ports have interrupt function and support level interrupt and double edge interrupt. The interrupt type is configured by the interrupt type register (GPIO_INT_TYPE_LEVEL) and interrupt priority register (GPIO_INT_ POLARITY). In order to use interrupts, the port must be configured as input mode.

12 Timer

12.1 TimerCCR

TimerCCR is a versatile timer with Capture and Reload capabilities. It can be used for the capture of digital signal events of various frequencies and duty cycles. The TimerCCR overload value is always from TIMER_RELOAD. There are three ways to trigger reloads, which are controlled by TIMER_RELOAD_SEL:

- Mode 1: software overloading;
- Mode 2: overflow automatic overload;
- Mode 3: only in the falling edge of the tccr_ex overloading;

TimerCCR functional modes include four modes, as follows:

1. Normal mode
As a normal Timer (TIMER_CLK_SEL = 1), the counter clock uses the input CLK /2, the initial value of the counter system is reset to 0, can be overloaded through the register TIMER_VALUE, when the counter overflows, an overflow interrupt is issued.

2. Event Timer mode
In this mode (TIMER_CLK_SEL is 2), the counter only accumulates on the falling edge of TCCR_EX, that is, it increases by 1 for each cycle of TCCR_EX, so it can be used to count the occurrence of events (TCCR_EX).

3. Gating Timer mode
In this mode (TIMER_CLK_SEL = 3), the counter accumulates only when TCCR_EX is high and stops at low level, so pulse width measurements can be made using this mode.

4. Capture patterns
Measure the pulse signal width, period, etc. (TCCR_EX).

12.2 Timer0

Timer0 has two uses, one is used as a normal timer, the other is used to generate PWM signal.

Here's how it works as a normal timer. When the timer is enabled, the setting value is reloaded into the timer, and then decremented by the clock (the clock frequency is optional) until the timer is 0, and an interrupt signal is given.

12.3 Timer1

Timer1 can only be used as a normal timer. When the timer is enabled, the setting value is reloaded into the timer, and then decremented by the clock (the clock frequency is optional) until the timer is 0, and an interrupt signal is given.

12.4 Timer2

Timer2 can only be used as a normal timer. When the timer is enabled, the setting value is reloaded into the timer, and then decremented by the clock (the clock frequency is optional) until the timer is 0, and an interrupt signal is given.

12.6 Core Timer

The system timer provides an easy way to use 24-bit cyclic decrement counter that starts working when the system timer is enabled. When the counter decrements to 0, the system timer makes an interrupt request to the vector interrupt controller to obtain a processor response and process the transaction of the system timer.

13 PWM

The PWM module uses Timer0 to generate the PWM. When Timer0 is used to generate the PWM, it is no longer used as a normal timer.

Key Features:

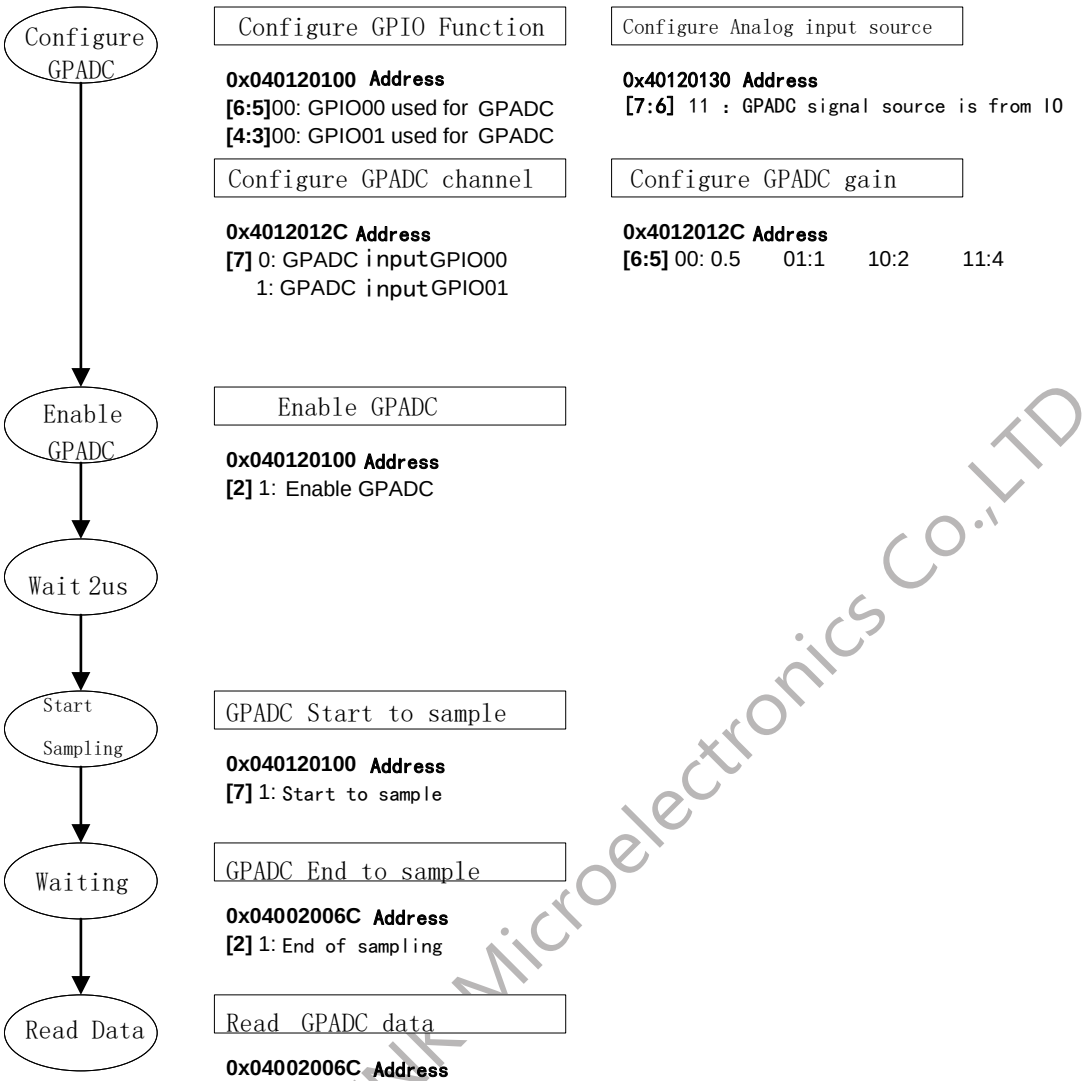
- Support up to 6 channel PWM signal;
- Each PWM has its own enabling switch, and each PWM has two modes: complementary mode and non-complementary mode. Each PWM route has its own dead zone control.
- The cycle of 6-channel PWM is controlled by the same timer. As for any 1-channel PWM, the frequency division can be carried out according to the cycle set by the timer (including multiple small cycles, which can be set as 1, 2, 4, 8 cycles).The reload value of the Timer should be set to an integer multiple of the minimum cycle of the PWM used (limited to 1, 2, 4, 8).

When Timer0 is used to control the PWM, the initial timer value is controlled by 0x40030000 and 0x40030008. These two registers are described in the Timer0 register description.

14 Other Peripherals

14.1 GPADC

GPADC can convert analog input signals into digital signals with 10bit sampling accuracy, which can be read by the MCU. The analog signal can be input by two IO chips, and the specific input IO can be configured through registers. The input range of the analog signal is determined by the set measurement channel range. It is also important to note that GPADC does not work together with radio frequency receivers.



14.2 Wakeup Trigger

The Wakeup Triger is used to wake up the CPU after sleep. You need to set the WAKE-UP-IO and the corresponding level.

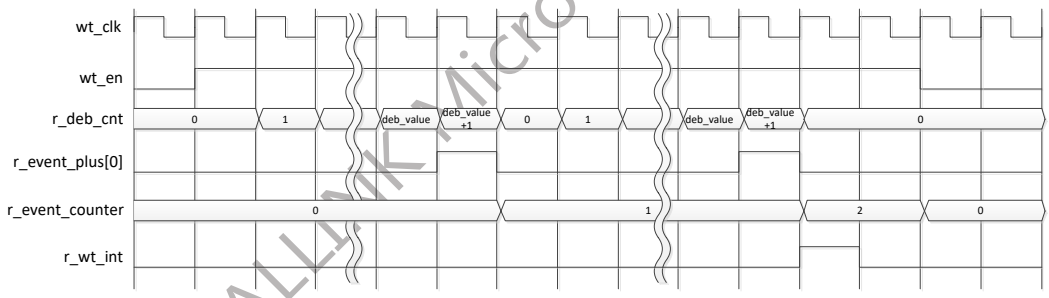
The wakeup trigger shall be used to wake up by low level, if high level is used to wake up, external pull-down resistance is required.

14.2.1 key features

- Wake-up level: The wake-up level can be set to high or low;
- Debounce time: A wake-up signal that remains unchanged for a DEB_VALUE time is considered a valid level signal, and a DEB_VALUE value can be matched.

14.2.2 description

The Debounce module has a built-in counter r_deb_value +1 (+1 allows for deb_value=0), which counts the enabled signal as WT_EN;When the WT_INT signal is high, the counter clears 0 (low power);When the WT_EN signal is set high, the counter starts to work. I_WT_IN signal is debounce according to WT_INPUT_EN, WT_POL. When the counter value is DEB_VALUE +1, a valid EVENT_PLUS for each route is generated and stored in R_EVENT_PLUS [31:0], while the counter clears 0. The following diagram shows a sequence diagram in one case:



wt_compare=2

Counter: Counts of r_event_plus. If the number of events is greater than or equal to (WT_Compare +1), an interrupt is generated.For example, if WT_Compare is set to 1, 2 events are required for an interrupt to occur. Note: If WT_Compare is set to 0, an interrupt is generated immediately and no event is required.

When the WT_EN signal is set to 0, all registers in the wakeup_timer module are set to reset, and the module is inactive.

14.3 Watchdog

Watch Dog Timer (WDT) is an on-chip RC oscillation Timer with self-oscillating mode. It does not need any peripheral components. The WDT maintains timing even when the chip's master clock stops working. It can be used to detect and resolve faults caused by software errors. System reset occurs when the WDT timing overflows.

14.3.1 key features

- Free running increment counter;
- The clock is provided by a separate RC oscillator (works in stop and sleep modes);
- When the WDT is activated, the counter interrupts when it counts to the set value and resets when it counts again to the set value.

14.3.2 description

Set the WDT timeout time in the load value register (WATCHDOG_COMP_VALUE_H/ WATCHDOG_COMP_VALUE_M/ WATCHDOG_COMP_VALUE_H), enable the latch register (WATCHDOG_COMP_VLD) (that is, write 1), then the load value is valid, open WATCHDOG; At this time, the counter begins to increment the count from the reset value (0). When the counter value is the load value, an interrupt signal will be generated. At this time, the counter value will be cleared to zero and re-counted, and then the reset signal will be reset when the count reaches the set value. Whenever you write 1 (Feed Dog) in the counter zero register before the reset, the counter value is reset to avoid a reset.

Watchdog uses 32kHz clock with 24 bit counter. The maximum setting time of WDT is 524s, and the overflow reset time is 2 times of the set time. The overflow reset time will be affected by ambient temperature, supply voltage and other parameters.

If the Watchdog overflow results in a reset, the WATCHDOG_FLAG bit of the state register (WATCHDOG_STATE) is set. The user can determine if the reset was caused by the Watchdog overflow.

Notes:

- If you use the Watchdog function, be sure to place a clear (feed dog) command somewhere in your program to ensure that the Watchdog can be cleared before it overflows. Otherwise, the chip will be reset non-stop, causing the system can’t work normally.
- Can’t clear Watchdog in interrupt program, otherwise you can’t detect the main program "running" situation.
- Procedures should have a clear Watchdog operation in the main program, try not to clear the Watchdog in multiple branches, this structure can maximize the protection function of the Watchdog counter.

- Watchdog counter different chip overflow time has certain difference. When setting the time to 'feed dog', its value should have significant redundancy with the overflow time to avoid an unnecessary reset of the Watchdog.

14.4 LVD

The AK801 chip supports Low voltage detection and can provide two functions.

- In the working mode of the chip, check the chip power supply voltage. When the power supply voltage is found to be lower than the preset threshold, LVD module will interrupt the CPU.
- In the chip sleep mode, detect the power supply voltage. When the chip power supply voltage is found to be higher than the threshold value, wake up the chip and enter the working state.

15 Data communication

15.1 Overview

- Support TX & RX data packet format of BLE 4.0
- Support 2.4G private protocol:
  a) 1~3 bytes preamble
  b) 3~5 bytes address
  c) Support the same header as the BLE format definition or software customization
  d) Support the same CRC and Whitening as the BLE format definition
- Support 64 bytes double buffer mode and 128 bytes single buffer mode
- Support two matching addresses
- After sending and receiving, the corresponding interrupt can be generated to the CPU

15.2 Working mode

The data communication buffer of AK801 can be configured as single buffer (128 bytes) working mode or double buffer (64 bytes) working mode. Specifically divided into the following categories:

1) Single Buffer transmission mode, up to 128 bytes of data can be sent.
2) Double buffer transmission mode.
3) Single buffer receiving mode, the maximum can receive 128 bytes of data. .
4) Double buffer receiving mode.
5) Double-buffered receiving and transmission mode.

In the receiving mode, the data length that can be received is divided into the following two situations:

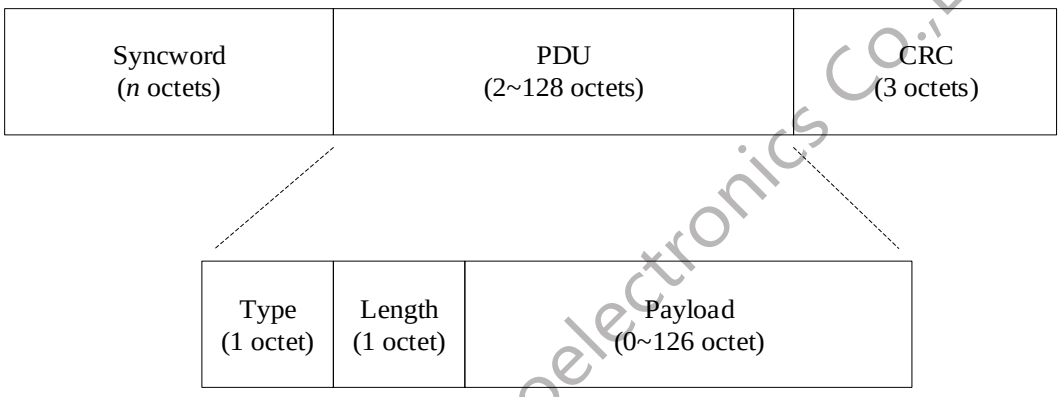
1) Automatic mode, the length uses the Length field in the PUD header of the received broadcast packet;
2) In manual mode, the length uses the value configured in the register.

In the sending mode, after the sending is started, the data buffer cannot be read or written by the CPU until the operation is completed (the corresponding status register is valid), otherwise unpredictable results will occur.

Similarly, in the receiving mode, after the receiving is started, the data buffer cannot be read or written by the CPU until the receiving is completed (the corresponding status register is valid), otherwise unpredictable results will occur.

To start the baseband, you need to configure the registers first, and then perform operations (read or write) on sending and receiving data. Most of the register settings can be configured with default values. The manual gives the registers that need to be modified in different working modes. The register Rx represents the register of offset address x.

15.3 Packet format



syncword: related codeword. If the Bluetooth standard is adopted, syncword = preamble + address, a total of 5 bytes; the private protocol can be set arbitrarily, and the number of bytes is variable.

PDU: Data segment. If the Bluetooth standard is adopted, PDU = Header (that is, Type+Length) + Payload. The private protocol is not distinguished. The Type and Length here are directly regarded as the payload. The required Type and Length in the standard specification are taken from The configuration register is read.

15.4 Bit sequence

Divided into bit order and byte order, follow the following rules.

- Multi-byte part, except for CRC and MIC (message integrity check, 8 bytes are added to the end of the data segment when encrypting the packet), the rest are in little endian order.
- Inside each byte, except for CRC, the low bit (LSb) in the byte is sent or received first.

For example: for broadcast address 32'0x8E89BED6, byte 8'0xD6 (8'b1101_0110) is transmitted or received first, and the byte is 0, 1, 1, 0, 1, 0, 1 in the order of bit transmission or reception.

15.5 Address Rules

When the address (the address part of the syncword) of the data channel (37/38/39 is the broadcast channel) meets the following specifications, the receiver can obtain better performance.

- Cannot exceed 6 consecutive 0s or consecutive 1s.

- It cannot be equal to the address of the broadcast channel.
- It cannot be different from the address of the broadcast channel by 1.
- There cannot be 4 octets exactly the same.
- No more than 24 transitions (change from 0 to 1 in the adjacent two bits in the address sequence, or change from 1 to 0 is counted as one transition).
- The high 6bit has at least two transitions.

This device complies with part 15 of the FCC rules. Operation is subject to the following two conditions:
(1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.
Warning: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.
NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed andused in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
—Reorient or relocate the receiving antenna.
—Increase the separation between the equipment and receiver.
—Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
—Consult the dealer or an experienced radio/TV technician for help.
If the FCC identification number is not visible when the module is installed inside another device, then the outside of the device into which the module is installed must also display a label referring to the enclosed module. This exterior label can use wording such as the following: "Contains Transmitter Module FCC ID: 2A7EM-A19BEACON Or Contains FCC ID: 2A7EM-A19BEACON "

2.2 List of applicable FCC rules
FCC CFR Title 47 Part 15 Subpart C Section 15.247
2.3 Summarize the specific operational use conditions
The module has been certified for Fix, Mobile, Portable applications. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
2.4 Limited module procedures
The module has not its own RF shielding, which belong to Limited module Standard requires: Clear and specific instructions describing the conditions, limitations and procedures for third-parties to use and/or integrate the module into a host device (see Comprehensive integration instructions below).

2.5 Trace antenna designs

Not applicable

2.6 RF exposure considerations

This modular complies with FCC RF radiation exposure limits set forth for an uncontrolled environment.
This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.
This modular must be installed and operated with a minimum distance of 20 cm between the radiator and user body.

2.7 Antennas

Integral antenna with antenna gain 1.1dBi, The antenna is permanently attached, can't be replaced.

2.8 Label and compliance information

This device complies with part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.
Warning: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.
NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates uses and can radiate radio frequency energy and, if not installed andused in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:
- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
-Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
-Consult the dealer or an experienced radio/TV technician for help.
If the FCC identification number is not visible when the module is installed inside another device, then the outside of the device into which the module is installed must also display a label referring to the enclosed module. This exterior label can use wording such as the following: "Contains Transmitter Module FCC ID: 2A7EM-A19BEACON Or Contains FCC ID: 2A7EM-A19BEACON "

2.9 Information on test modes and additional testing requirements

The modular transmitter is only FCC authorized for the specific rule parts (FCC Part 15.247) list on the grant, and that the host product manufacturer is responsible for compliance to any other FCC rules that apply to the host not covered by the modular transmitter grant of certification. The final host product still requires Part 15 Subpart B compliance testing with the modular transmitter installed when contains digital circuity.

2.10 Additional testing, Part 15 Subpart B disclaimer

When testing host product, the host manufacture should follow FCC KDB Publication 996369 D04 Module Integration Guide for testing the host products. The host manufacturer may operate their product during the measurements.