BS9000AM Series V1.0

General-purpose MCU With EEPROM, 12bit-ADC, LIN Communication

V1.0 2022-05-30

Table of Contents

1 BS9000AM Series MCU Overview

1.1 Features

- Core: High-speed 8051, 1T Instruction cycle
  - Operating Frequency: 24MHz /16MHz /12MHz/ 8MHz
  - Clock Offset: ±1% @-20℃~65℃, 5V or ±3% @-40℃ ~125℃, 5V
- Memories
  - 30.5KBytes FLASH
  - 1024Bytes EEPROM
  - 256 Bytes(idata)+2048 Bytes(xdata) SRAM
  - Support IAP online update, memories protection
- Clock Source, Reset and Power Management
  - Internal Low-speed RC: 32768Hz, clock offset within ±3%@25℃, 5V or ±5% @-40℃ ~125℃, 5V
  - Internal High-speed OSC: 1MHz
  - External OSC: 16MHz/12MHz/8MHz
  - 7 Reset modes, among which the BOR mode is ON with the voltage selectable 2.6V/2.8V/3.7V/4.2V
  - Low voltage detection: 3V/3.3V/3.6V/3.9V/4.5V
- IO
  - 10k Embedded Pull-up/ Pull-down Resistors
  - Provide multiplexing of Peripheral functions
  - I/O Lines with External Interrupt Capability, INT0 to 2 enable Rising Edge, Falling Edge or both interrupt respectively, INT3 enables both rising and falling interrupt.
- Communications
  - UART
  - IIC slave communication supporting 100/400kHz
  - SPI: master with a frequency up to 4M and 2M slave
  - SCI: LIN2.1 protocol, up to 115200bps Baud
- 16-bit PWM
  - PWM0 supports 4-channel input/output with 3-channel complementary output
  - PWM1 supports 3-channel input/output
- Operating voltage: 3.0V to 5.5V, 3.3V or 5.0V typ
- Operating temperature: -40℃ to 125℃
- High accuracy 12-bit ADC
  - Up to 24 Analog Input channels
  - 2.048V/4.096V internal reference voltage detection
- Interrupt
  - High/Low-Level Interrupt Selection
  - ADC, CSD, INT0/1/2/3, LVDT, Timer0/1/2, RTC, WDT, UART, SCI, IIC, SPI
- Timer
  - 3 16-bit Timer0/1/2
  - 16-bit RTC, wake-up in low-power mode
  - Watchdog Timer
- Low-power Mode
  - Wait Mode
  - Sleep Mode, 20μA @5V typ consumption
- TouchKey
  - The sensitivity of each key can be set individually
  - Capacitive TouchKey can be multiplexed with GPIO lines
- 2-wire programming and single-wire debugging emulator IO
- Packages: QFN20/TSSOP28

1.2 Description

BS9000AM series are 8-bit general-purpose microcontrollers that meets the quality requirements of AECQ0-100 Grade 1. Based on 8051 CORE and 1T instruction cycle, it runs faster than the Standard 8051 with 12T. Also, it is compatible with Standard 8051 instructions.

The peripherals of BS9000AM series devices includes WDT, PWM0/1, Timer0, Timer1, Timer2, RTC, IIC, SPI, UART, SCI with Lin2.1 protocol, LV Detection, BOR, 12-bit successive-approximation ADC,Touch-key Detection and Low-power mode. The memory supports EEPROM real-time data storage. The BS9000AM series are available in QFN20 and TSSOP28 packages. The BS9000AMxx is suitable for many applications such as end actuator unit, end sensor detection control.

1.3 System Block Diagram

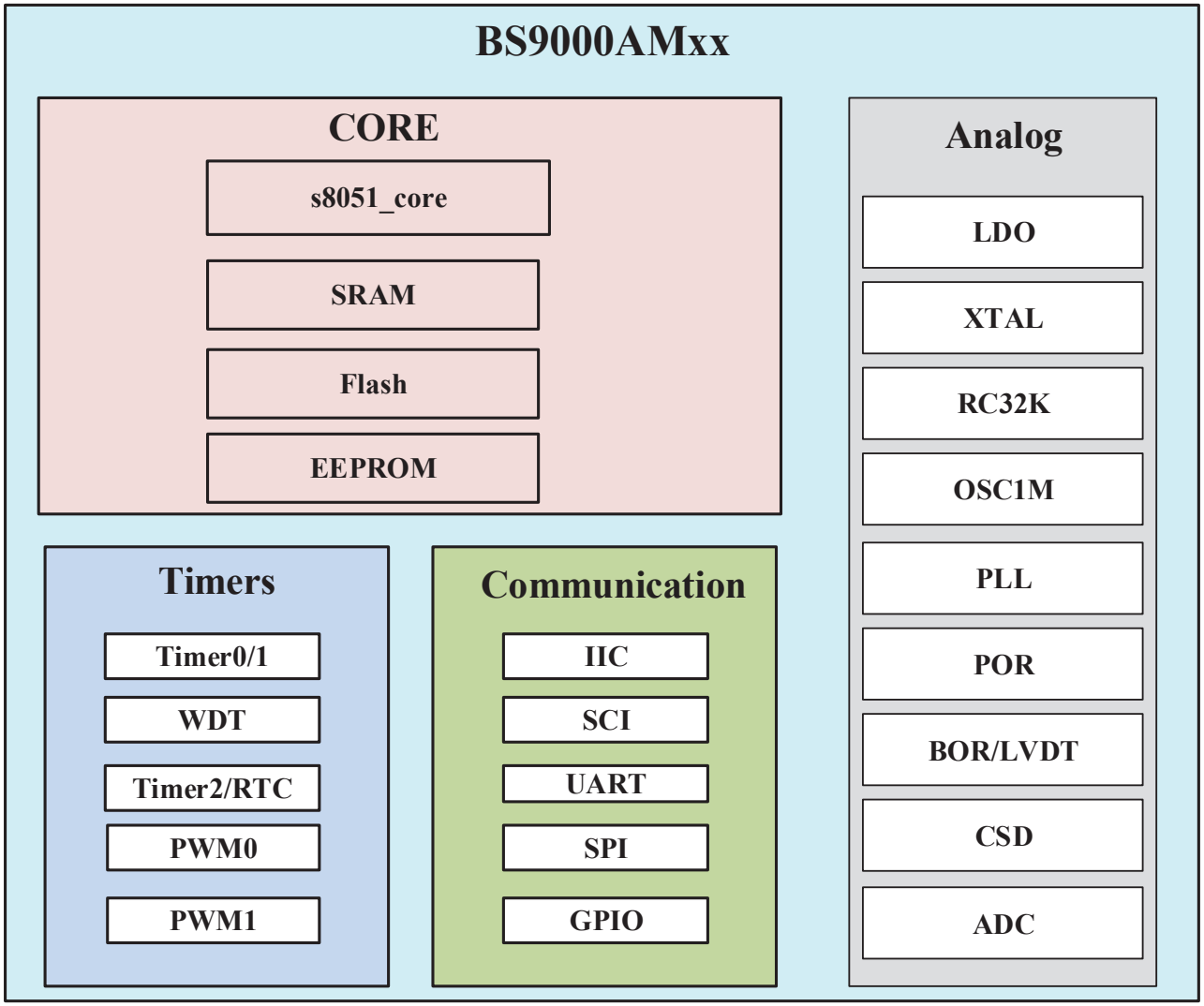


Figure 0.1 BS9000AMxx Block Diagram

1.4 Clock Diagram

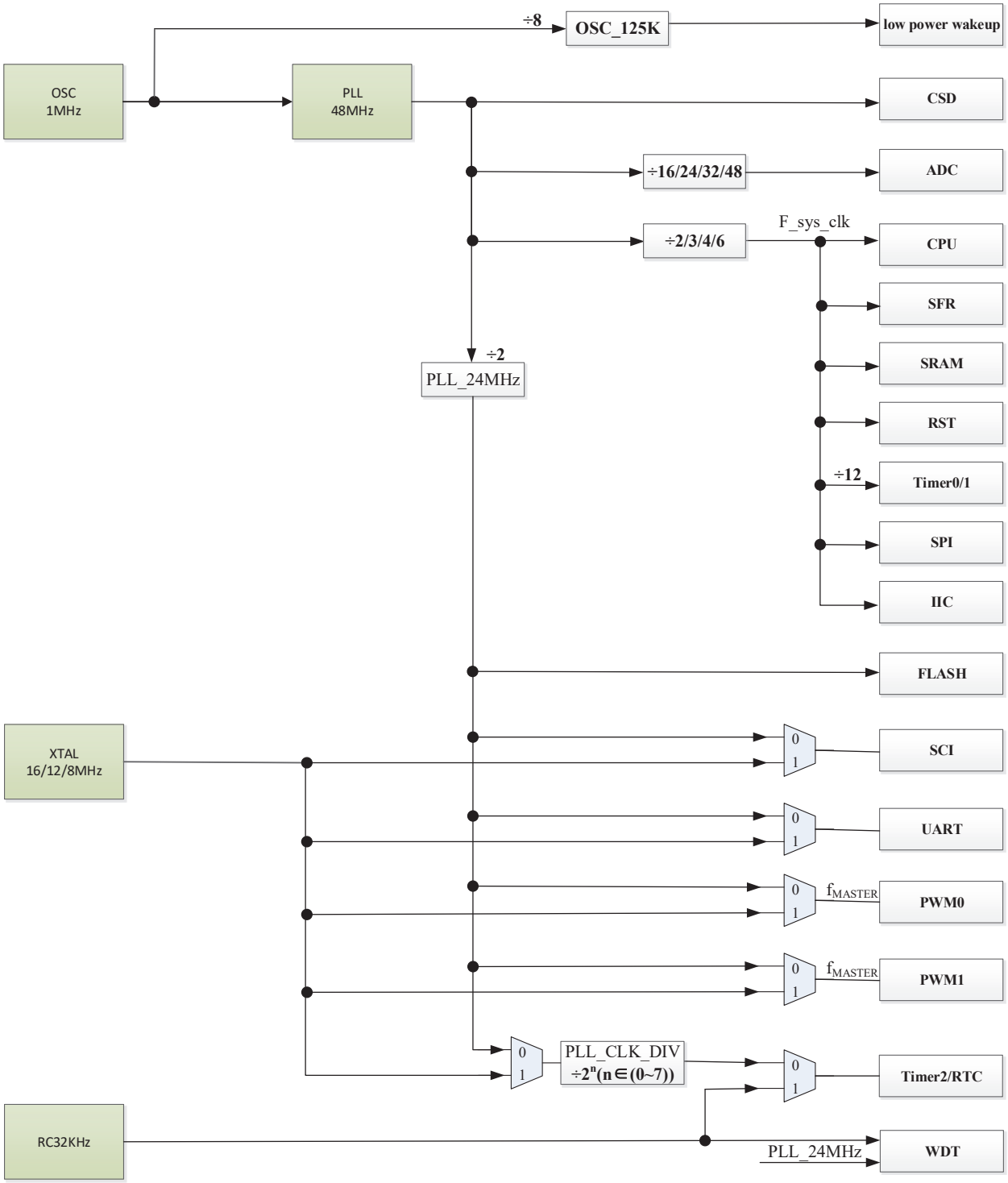


Figure 0.2 Clock Diagram

Main Clock

- OSC_125KHz: a timer clock used in Low-consumption Wakeup Interrupt Mode.
- F_sys_clk: select clock frequency of 24MHz/16MHz/12MHz/8MHz.

- $f_{MASTER:}$ PWM PLL_24MHz/XTAL can be selected.
- PLL_24MHz: a clock for FLASH program and read, SCI/UART/PWM0/PWM1.
- XTAL: External Crystal. When the frequency is at 16M or 12M or 8M, it acts as a timer for Timer2/RTC, also a clock SCI/UART/ PWM0/PWM1.
- RC32KHz: The frequency is 32768Hz, used for Timer2, RTC and WDT setting.

1.5 Selection Table

P/N	BS9000AM20-QBBX	BS9000AM28-TBBX
Operating Voltage	3.0V to 5.5V	3.0V to 5.5V
Core	1T 8051	1T 8051
Operating Frequency	24M	24M
FLASH (Byte)	30.5K	30.5K
SRAM (Byte)	256+2048	256+2048
EEPROM (Byte)	1024	1024
GPIO	18	26
ADC	16	24
KEY	16	24
Timer	4	4
PWM0	4ch	4ch
PWM1	3ch	3ch
SPI	1	1
INT	18	26
IIC	1	1
UART	1	1
SCI	1	1
Package	QFN20	TSSOP28

Table 0.3 Selection Table

1.6 Pin configuration

1.6.1 QFN20

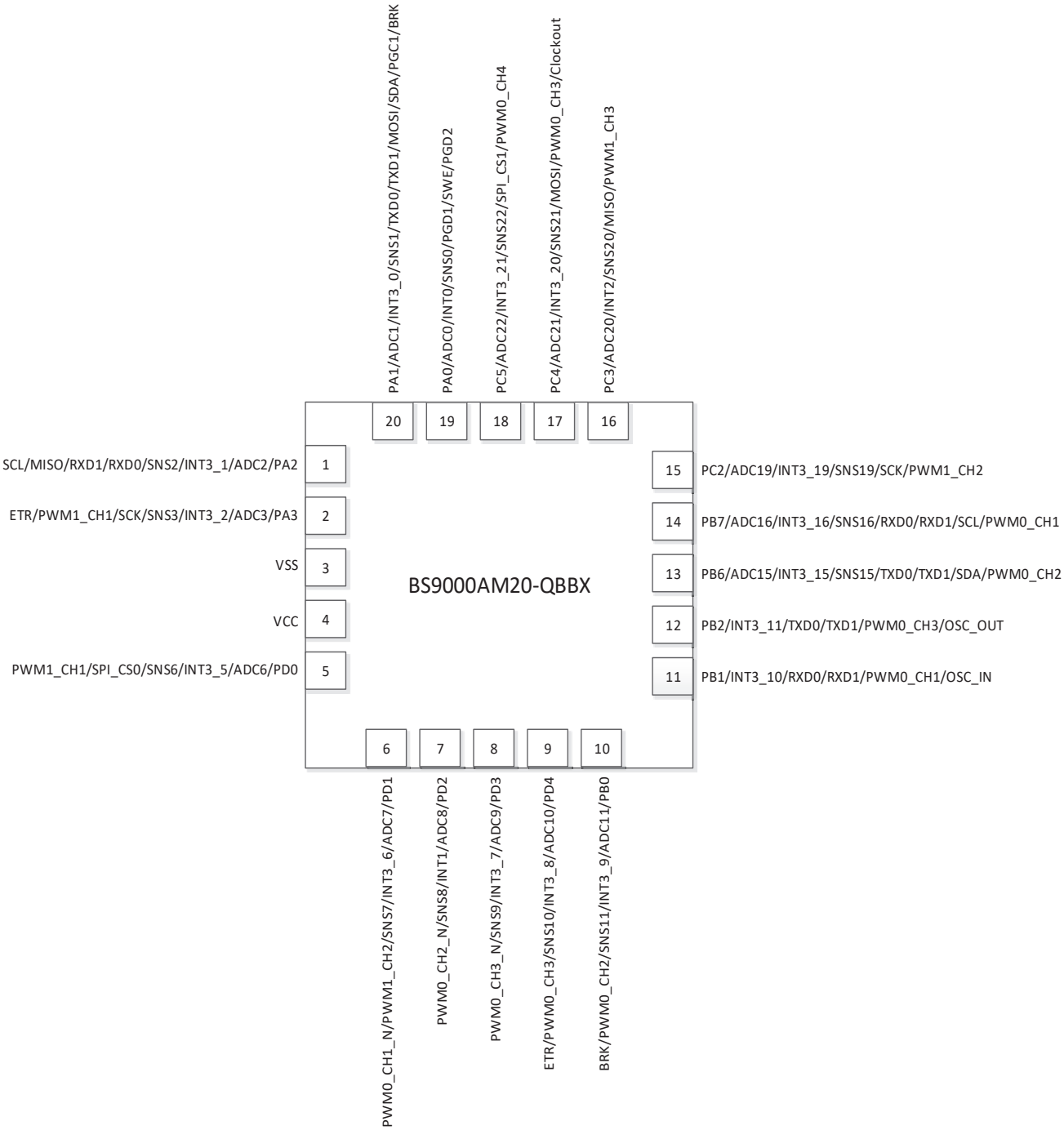


Figure 0.4 BS9000AM20-QBBX Package and Pinout Drawing

1.6.2 TSSOP28

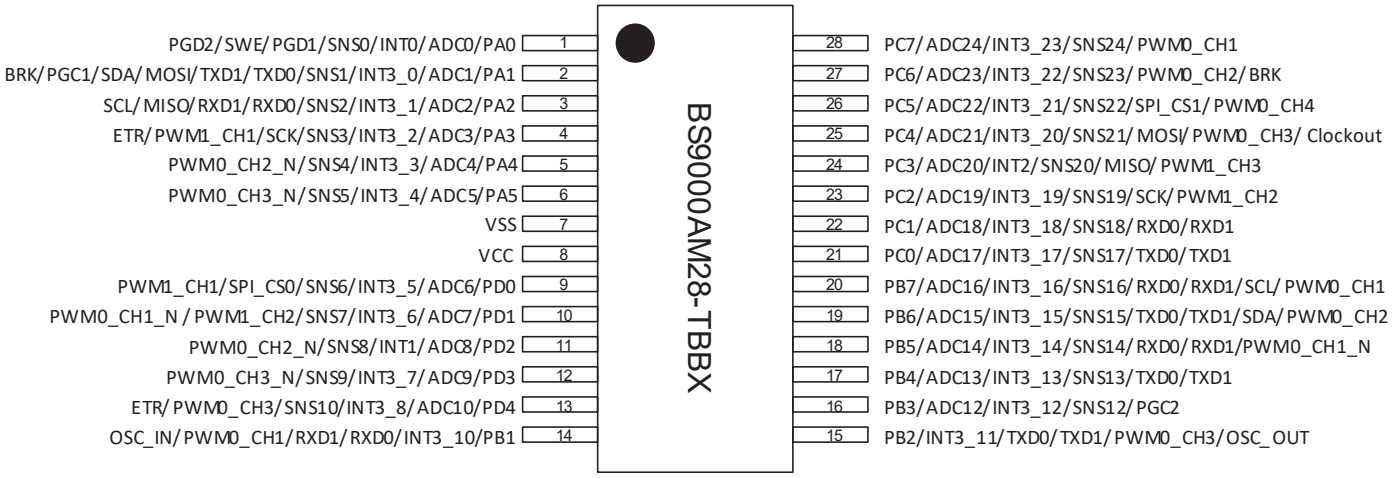


Figure 0.5 BS9000AM28-TBBX Package and Pinout Drawing

1.7 Pin Functions

BS9000AM20-xxxx	BS9000AM28-xxxx	Function
19	1	By Default: GPIO <PA0> Other functions: ADC00: ADC channel 0 INT0: External Interrupt 0 PGD1: Programming Interface PGD2: Programming Interface SWE: Single Wire Emulation IO SNS0: TouchKey channel 0
20	2	By Default: GPIO <PA1> Other functions: ADC01: ADC channel 1 INT3_0: External Interrupt 3_0 TXD0: Transmit Serial Data TXD1: Transmit Serial Data MOSI: Master Out Slave In in SPI Master Mode SDA：IIC serial data line PGC1: Programming Interface BRK: Break input SNS1: TouchKey channel 1
1	3	By Default: GPIO <PA2> Other functions: ADC02: ADC channel 2 INT3_1: External Interrupt 3_1 RXD0: Receive Serial Data RXD1: Receive Serial Data MISO: Master In Slave Out (MISO) in SPI Master mode SCL: IIC serial clock line SNS2: TouchKey channel 2
2	4	By Default: GPIO <PA3> Other functions: ADC03: ADC channel 3 INT3_2: External Interrupt 3_2 SCK: SPI clock line PWM1_CH1：PWM1 channel ETR：External interrupt SNS3：TouchKey channel 3

	5	By default: GPIO <PA4> Other functions: ADC04: ADC channel 4 INT3_3: External interrupt 3_3 PWM0_CH1_N: PWM0 complementary output channel SNS4: TouchKey channel 4
	6	By default: GPIO <PA5> Other functions: ADC05: ADC channel 5 INT3_4: external interrupt 3_4 PWM0_CH2_N: PWM0 complementary output channel SNS5: TouchKey channel 5
3	7	By default: Ground <VSS>
4	8	By default: Power Supply <VCC>
5	9	By default: GPIO <PD0> Other functions: ADC6: ADC channel 6 INT3_5: external interrupt 3_5 SPI_CS0: SPI chip select signal PWM1_CH1: PWM1 channel SNS6: TouchKey channel 6
6	10	By default: GPIO <PD1> Other functions: ADC7: ADC channel 7 INT3_6: external interrupt 3_6 PWM1_CH1: PWM 1channel PWM0_CH1_N: PWM0 complementary output channel SNS7: TouchKey channel 7
7	11	By default: GPIO <PD2> Other functions: ADC8: ADC channel 8 INT1: external interrupt 1 PWM0_CH1_N: PWM0 complementary output channel SNS8: TouchKey channel 8
8	12	By default: GPIO <PD3> Other functions: ADC9: ADC channel 9 INT3_7: external interrupt 3_7 PWM0_CH2_N：PWM0 complementary output channel SNS9: TouchKey channel 9
9	13	By default: GPIO <PD4> Other functions: ADC10: ADC channel 10 INT3_8: external interrupt 3_8 PWM0_CH2: PWM0 channel ETR: external interrupt SNS10: TouchKey channel 10

10		By default: GPIO <PB0> Other functions: ADC11: ADC channel 11 INT3_9: external interrupt 3_9 PWM0_CH1: PWM0 channel BRK: Break output SNS11: TouchKey channel 11
11	14	By default: GPIO <PB1> Other functions: INT3_10: external interrupt 3_10 RXD0: Receive Serial Data RXD1：Receive Serial Data PWM0_CH1: PWM0 channel OSC_IN: external oscillator input
12	15	By default: GPIO <PB2> Other functions: INT3_11: External interrupt 3_11 TXD0: Transmit Serial Data TXD1: Transmit Serial Data PWM0_CH2: PWM0 channel OSC_OUT: external oscillator output
	16	By default: GPIO <PB3> Other functions: ADC12: ADC channel 12 INT3_12: external interrupt 3_12 PGC2: programming interface SNS12: TouchKey channel 12
	17	By default: GPIO <PB4> Other functions: ADC13: ADC channel 13 INT3_13: external interrupt 3_13 TXD0: Transmit Serial Data TXD1: Transmit Serial Data SNS13: TouchKey channel 13
	18	By default: GPIO <PB5> Other functions: ADC14: ADC channel 14 INT3_14: external interrupt 3_14 RXD0: Receive Serial Data RXD1: Receive Serial Data PWM0_CH1_N: PWM complementary output channel SNS14: TouchKey channel 14
13	19	By default: GPIO <PB6> Other functions: ADC15: ADC channel 15 INT3_15: external interrupt 3_15 TXD0: Transmit Serial Data TXD1: Transmit Serial Data

		SDA: IIC serial data line PWM0_CH1: PWM0 channel SNS15: TouchKey channel 15
14	20	By default: GPIO <PB7> Other functions: ADC16: ADC channel 16 INT3_16: external interrupt 3_16 RXD0: Receive Serial Data RXD1: Receive Serial Data SCL: IIC serial clock line PWM0_CH1: PWM0 channel SNS16: TouchKey channel 16
	21	By default: GPIO <PC0> Other functions: ADC17: ADC channel 17 INT3_17: external channel 3_17 TXD0: Transmit Serial Data TXD1: Transmit Serial Data SNS17: TouchKey channel 17
	22	By default: GPIO <PC1> Other functions: ADC18: ADC channel 18 INT3_18: external interrupt 3_18 RXD0: Receive Serial Data RXD1: Receive Serial Data SNS18: Touch key channel 18
15	23	By default: GPIO <PC2> Other functions: ADC19: ADC channel 19 INT3_19: external interrupt 3_19 SCK: SPI serial clock line PWM1_CH1: PWM1 channel SNS19: TouchKey channel 19
16	24	By default: GPIO <PC3> Other functions: ADC20: ADC channel 20 INT2: external interrupt 2 MISO: Master In Slave Out (MISO) in SPI Master mode PWM1_CH2：PWM1 channel SNS20: TouchKey channel 20
17	25	By default: GPIO <PC4> Other functions: ADC21: ADC channel 21 INT3_20: external interrupt 3_20 MOSI: Master Out Slave In in SPI Master Mode PWM0_CH2: PWM0 channel SNS21: TouchKey channel 21

18	26	By default: GPIO <PC5> Other functions: ADC22: ADC channel 22 INT3_21: external interrupt 3_21 SPI_CS1: chip select PWM0_CH3: PWM0 channel SNS22: TouchKey channel 22
	27	By default: GPIO <PC6> Other functions: ADC23: ADC channel 23 INT3_22: external interrupt 3_22 PWM0_CH1: PWM0 channel BRK: break input SNS23: TouchKey channel 23
	28	By default: GPIO <PC7> Other functions: ADC24: ADC channel 24 INT3_23: external interrupt 3_23 PWM0_CH1: PWM0 channel SNS24: TouchKey channel 24

Table 0.6 PIN MAP

Note : Oscillator uses the OSC_IN pin to input clock.

2 Electrical Characteristics

2.1 AC Characteristics

Parameter	Symbol	Conditions	Clock Skew	Unit
Fundamental frequency	OSC1M	Ta=25℃, @5V	±1%	MHz
		Ta= -40℃ ~ 85℃, @5V	±2%	
		Ta= -40℃ ~ 125℃, @5V	±3%	
		VCC=3.0V ~ 5.5V, environment temperature 25℃	±3%	
System clock	F_sys_clk	Ta=25℃ @5V	±1%	
		Ta= -40℃ ~ 85℃, @5V	±2%	
		Ta= -40℃ to 125℃, @5V	±3%	
		VCC=3.0V ~ 5.5V, environment temperature 25℃	±3%	
Internal low-speed clock	RC32k	Ta=25℃, @5V	±3%	kHz
		Ta= -40℃ to 125℃, @5V	±5%	
		VCC=3.0V ~ 5.5V, environment temperature 25℃	±5%	

Table 0.1 AC characteristics

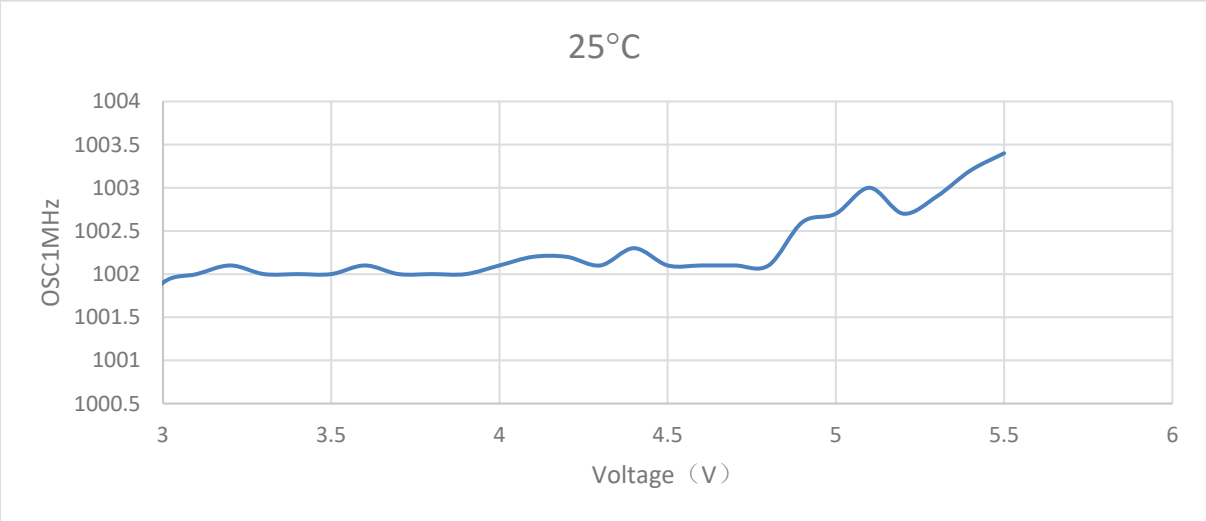


Figure 0.2 OSC1MHz Voltage Curve

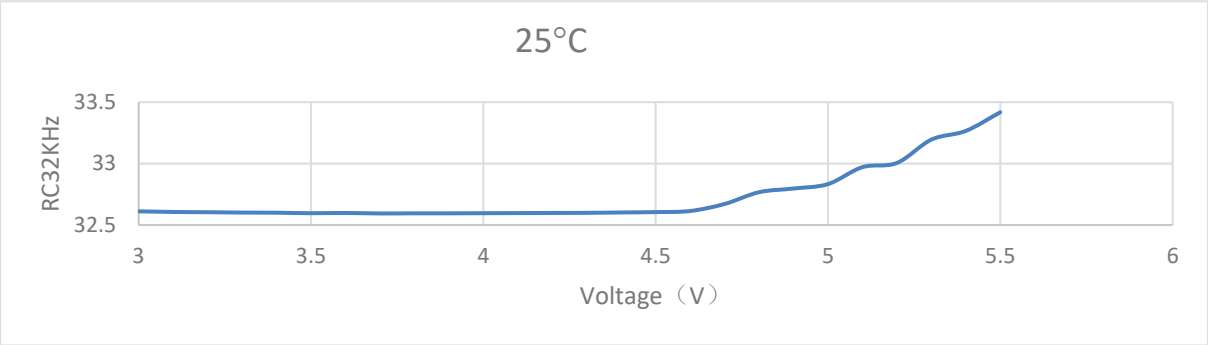


Figure 0.3 RC32KHz Voltage Curve Graph

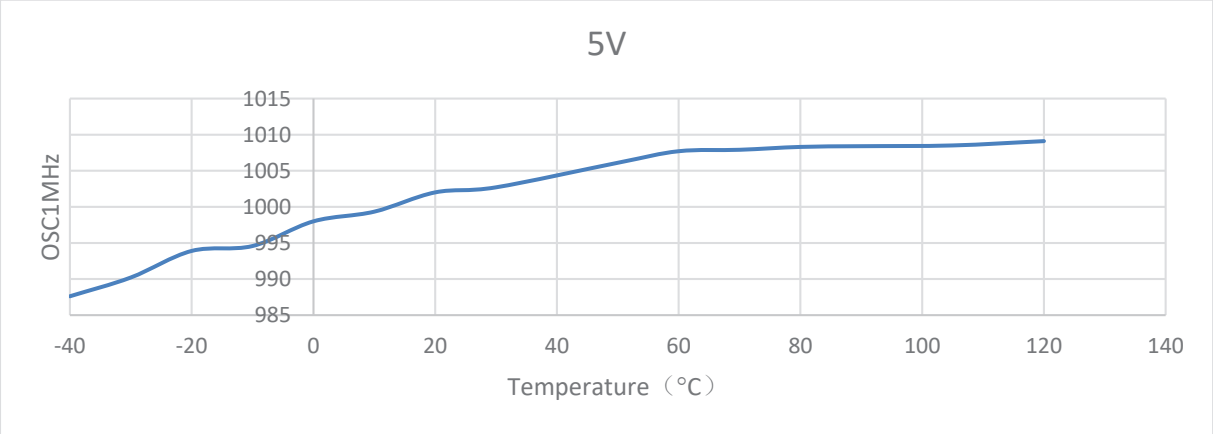


Figure 0.4 OSC1MHz Temperature Curve Graph

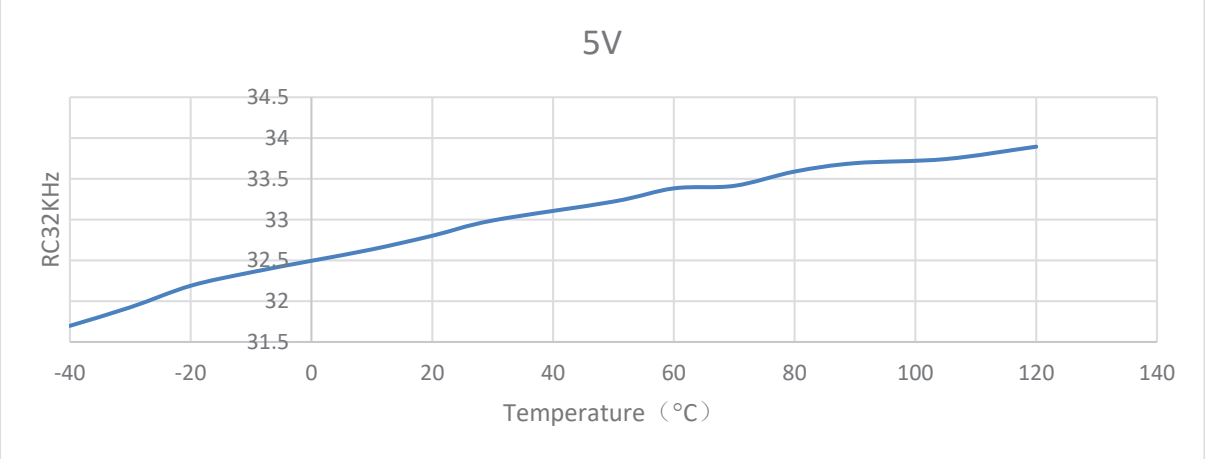


Figure 0.5 RC32KHz Temperature Curve Graph

2.2 DC Characteristics

Table 2.6 DC characteristics (Tamb = 25℃, unless otherwise specified)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Operating voltage	VCC	-	3.0	-	5.5	V

Operating Modes	Active	@5V, 24M system clock without load, other functions turned off	-	9.97		mA
		@5V, 16M, system clock without load, other functions turned off	-	7.40		mA
		@5V, 12M system clock without load, other functions turned off	-	5.90		mA
		@5V, 8M system clock without load, other functions turned off	-	4.60		mA
		@3.3V, 24M system clock without load, other functions turned off	-	9.97	-	mA
		@3.3V, 16M system clock without load, other functions turned off	-	7.40	-	mA
		@3.3V, 12M system clock without load, other functions turned off	-	5.90	-	mA
		@3.3V, 8M system clock without load, other functions turned off	-	4.57	-	mA
		@3.0V, 24M system clock without load, other functions turned off	-	9.97	-	mA
		@3.0V, 16M system clock without load, other functions turned off	-	7.40	-	mA
		@3.0V, 12M system clock without load, other functions turned off	-	5.90	-	uA
		@3.0V, 8M system clock without load, other functions turned off	-	4.57	-	uA
	Wait	@5V 24M system clock, IO output low, enter wait mode, other functions turned off	-	3.73	-	mA
		@3.3V 24M system clock, IO output low, enter wait mode, other functions turned off	-	3.73	-	mA
		@3.0V 24M system clock, IO output low, enter wait mode, other functions turned off	-	3.73	-	mA
		@5V 16M system clock, IO output low, enter wait mode, other functions turned off	-	3.77	-	mA
		@3.3V 16M system clock, IO output low, enter wait mode, other functions turned off	-	3.70	-	mA
		@3.0V 16M system clock, IO output low, enter wait mode, other functions turned off	-	3.70	-	mA
		@5V 12M system clock, IO output low, enter wait mode, other functions turned off	-	3.77	-	mA
		@3.3V 12M system clock, IO output low, enter wait mode, other functions turned off	-	3.73	-	mA
		@3.0V 12M system clock, IO output low, enter wait mode, other functions turned off	-	3.73	-	mA

		@5V 8M system clock, IO output low, enter wait mode, other functions turned off	-	3.73	-	mA
		@3.3V 8M system clock, IO output low, enter wait mode, other functions turned off	-	3.70	-	mA
		@3.0V 8M system clock, IO output low, enter wait mode, other functions turned off	-	3.67	-	mA
	Sleep	@5V 24M system clock, PCON = 0x01, IO output low,other functions turned off	-	21.33	-	μA
		@3.3V 24Msystem clock, PCON = 0x01, IO output low,other functions turned off	-	19.33	-	uA
		@3.0V 24Msystem clock, PCON = 0x01, IO output low, other functions turned off	-	21.00	-	μA
CSD parallel mode	Idle	@5V, CSD parallel mode, WDT interrupt wakes up after 2s, 2ms working time, IO output low, other functions turned off	-	36.07	-	μA
		@3.3V, CSD parallel mode, WDT interrupt wakes up after 2s, 2ms working time, IO output low, other functions turned off	-	33.70	-	μA
		@3.0V, CSD parallel mode, WDT interrupt wakes up after 2s, 2ms working time, IO output low, other functions turned off	-	35.10	-	μA
		@5.0V, CSD parallel mode, RTC wakes up after 2s interrupt, 2ms working time, IO output low, other functions turned off	-	765.33	-	μA
		@3.3V, CSD parallel mode, RTC wakes up after 2s interrupt, 2ms working time, IO output low, other functions turned off	-	754.67	-	μA
		@3.0V, CSD parallel mode, RTC wakes up after 2s interrupt, 2ms working time, IO output low, other functions turned off	-	756.33	-	μA
Input low voltage	V_{IL}	VCC=3.0~5.5V	-	-	0.3*VCC	V
Input high voltage	V_{IH}	VCC=3.0~5.5V	0.7*VCC	-	-	V
INT0/1/2/3 Input low voltage	V_{INTL}	VCC=3.0~5.5V	-	-	0.3*VCC	V
INT0/1/2/3 Input high voltage	V_{INTH}	VCC=3.0~5.5V	0.7*VCC	-	-	V
Output low voltage	V_{OL}	IOL=50mA@VCC=5V	-	-	0.1*VCC	V

Output high voltage	V_{OH}	IOH=15mA@VCC=5V	0.9VCC	-	-	V
IO sink current	IOL	V_{OL} =0.1VCC，@VCC=5V	-	50	-	mA
IO source current	IOH	V_{OH} =0.9VCC，@VCC=5V	-	15	-	mA
Input Leakage current	I_{Le}	VCC=5V	-	1	-	μA
Pull-up resistor	R_{P_u}	VCC=5V	-	10	-	kΩ
Pull-down resistor	R_{P_d}	VCC=5V	-	10	-	kΩ
ADC Operating current	I_{ADC}	@5V, 24M system clock without load, IO output low, enable ADC and turn on one channel, GET_ADC scan, turn off other functions	-	1.6	-	mA
LVDT Operating current	I_{LVDT}	@5V, 24M system clock without load; in sleep mode, IO output low, enable LVDT, turn off other functions	-	0.6	-	μA
CSD Operating current	I_{CSD}	@5V 24M system clock without load, IO output low, enable 6 channels of CSD and Timer0, turn off other functions	-	1.2	-	mA
PWM0 Operating current	I_{PWM0}	@5V 24M system clock without load, IO output low, enable PWM0, initate OC1 channel to 4K, turn off other functions	-	1.0	-	mA
PWM1 Operating current	I_{PWM1}	@5V 24M system clock without load, IO output low, enable PWM1, initate OC1 channel to 4K, turn off other functions	-	0.5	-	mA
EEPROM Erase current	$I_{\_E}$	@5V 24M system clock without load, IO output low, erase EEPROM in While, turn off other functions	-	3.3	-	mA
EEPROM Write current	$I_{\_W}$	@5V 24M system clock without load, IO output low, write a byte in While, turn off other functions	-	3.9	-	mA

2.3 ADC characteristics

Table 0.7 ADC characteristics (T_{amb} = 25℃,unless otherwise specified)

ADC electrical characteristics VDD=Vmin-5.5V, GND=0V, TA=+25℃						
Parameter	Symbol	Conditions	Min	Typ	Max	Unit
Supply voltage	V_{AD}	-	3.0	-	5.5	V
Accuracy	N_R	-	-	9		Bit
A/D input voltage	VAIN	-	VSS	-	V_{REF}	V
A/D input resistance	R_{AIN}	VCC=5V, RC filter	-	12	-	kΩ
		VCC=5V, NO RC filter	-	2.3	-	kΩ
A/D operating current	I_{AD}	-	-	0.9	-	mA
A/D input current	I_{ADIN}	-	-	-	1	μA
Differential Non-linearity error	D_{LE}	VDD=5.0V	-	±4	-	LSB
Integral Non-linearity error	L_{LE}	VDD=5.0V	-	±4	-	LSB
ADC sampling time	T_{AD}	-	1.33	-	1024	μs
ADC conversion time	TCON	-	7.75	-	1079	μs
Resolution	ADCRESO	-		12		Bit
Input channel	-	-	-	-	25	Channel

2.4 Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Unit
Supply voltage	VCC	VSS+3.0	-	VSS+5.5	V
Storage temperature	Tstg	-55	-	150	℃
Operating temperature	Totg	-40	-	125	℃
I/O input voltage	Vin	VSS-0.5	-	VCC+0.5	V
IOL total current	IOLA	-	100	-	mA
IOH total current	IOHA	-	-100	-	mA
I/O ESD voltage	ESD(HBM)	-8	-	8	kV
LATCH UP @T_A=25℃	I Trigger	-200		200	mA

Table 0.6 Absolute maximum ratings

Note: Stresses beyond those listed absolute maximum ratings may cause permanent damage to the device. The functional operation of the device may not be predicted beyond those indicated under recommended operating conditions. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

3 Memories

3.1 FLASH

Consisted of one block of 32KB and one block of 512bytes, the flash of BS9000AMxx can erase and write 10k times, and data is not lost after the brown-out. The allocation inside the Flash is as follows:

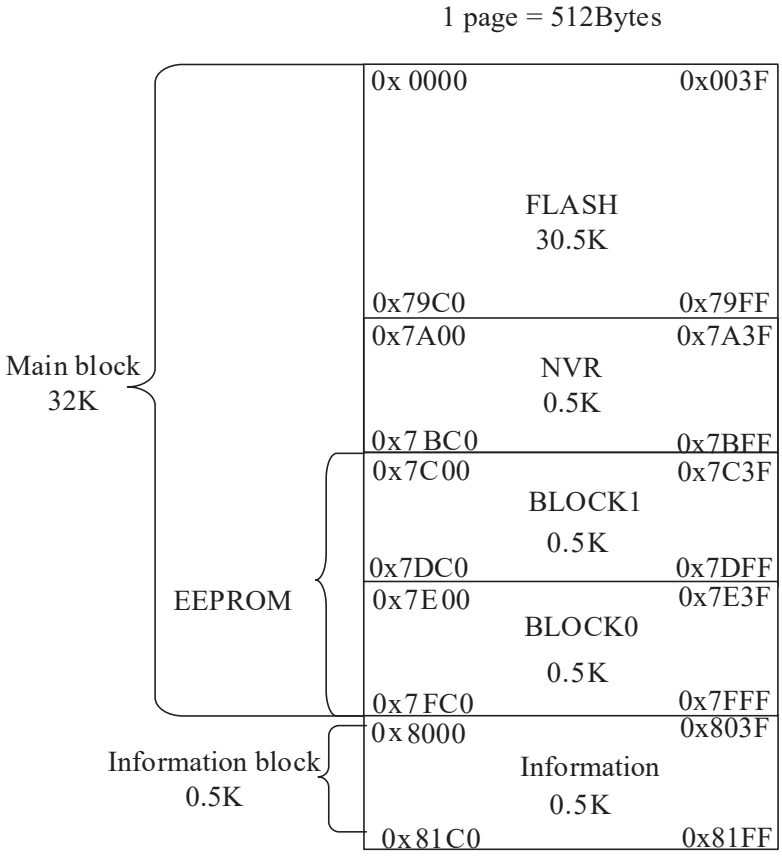


Figure 0.1 FLASH IP Memory Allocation

The flash has two blocks, which is Main block and information block, The last 1K of the main storage block is used for EEPROM functionality. The Main block can store the IC execution program. The information block is used for the storage of the configuration word information.

FLASH serves to solidify the user's programme and store the execution program.

NVR (KEY sector) is invisible to CPU and can not be written and read. This page can not be erased when the read protection is valid.

EEPROM area, can support CPU read and erase at any time, and is not protected by Flash read and write control.

The size of each page is 512 Bytes. The main block has 64 pages and the information block has 1 page.

FLASH Features is as follows:

- The Flash program uses 2-wire programming by multiplexing the PA[0](PGC) and PA[1](PGD) with the programming frequency ranging from 100K to 5M (allowing an error of 10%) and supports batch programming mode.
- Supports read protection and configuration word write protection
- BOOT program: it can be configured to disable Write protection , erase the Main block page and performs the byte write operation in the user code.
- Read out the code information stored in Flash when CPU is at working.
- Address overflow flag

3.2 FLASH Read and Write Protection

3.2.1 FLASH Write Protection

Write protection is configured by CFG_WPROTECT_PAGE. By default, the flash space protection is enabled. See Write Protection Control Range below.

Table 0.2 Write Protection Mapping

WPROTECT_PAGE	Write Protection Range
0	Not protected
1	NVR+0x0000-0x01FF
2	NVR+0x0000-0x03FF
3	NVR+0x0000-0x05FF
4	NVR+0x0000-0x07FF
5	NVR+0x0000-0x09FF
6	NVR+0x0000-0x0BFF
…	…
60	NVR+0x0000-0x77FF
61/62/63	NVR+0x0000-0x79FF

Note:

1. Write protection is solely used for CPU erase and write FLASH and do not limit program mode.
2. EEPROM sector supports CPU erase and write operation at any time and is without SFR write protection control.

3. INFORMATION sector does not support CPU erase and write operation at any time and is without Write protection control.
4. KEY NVR sector do not support CPU erase and write operation at any time and is without Write protection control. Note that 32 bytes from 0x7be0 to 0x7bff do not open to CPU operations and CPU can not read.

3.2.2 FLASH Read Protection

FLASH provides read protection functions including read protection deadlock and read protection encryption. The difference between them is that the latter can temporarily disable read protection by sending a decryption command in programming mode, while the former cannot. See the specific FLASH mapping below.

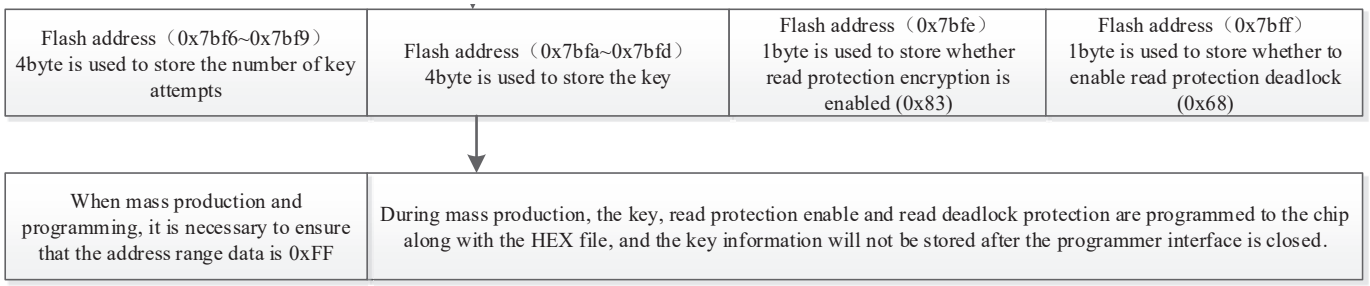


Figure 3.3 Read Protection

See below the Read Protection Control Flow Chart in programming mode.

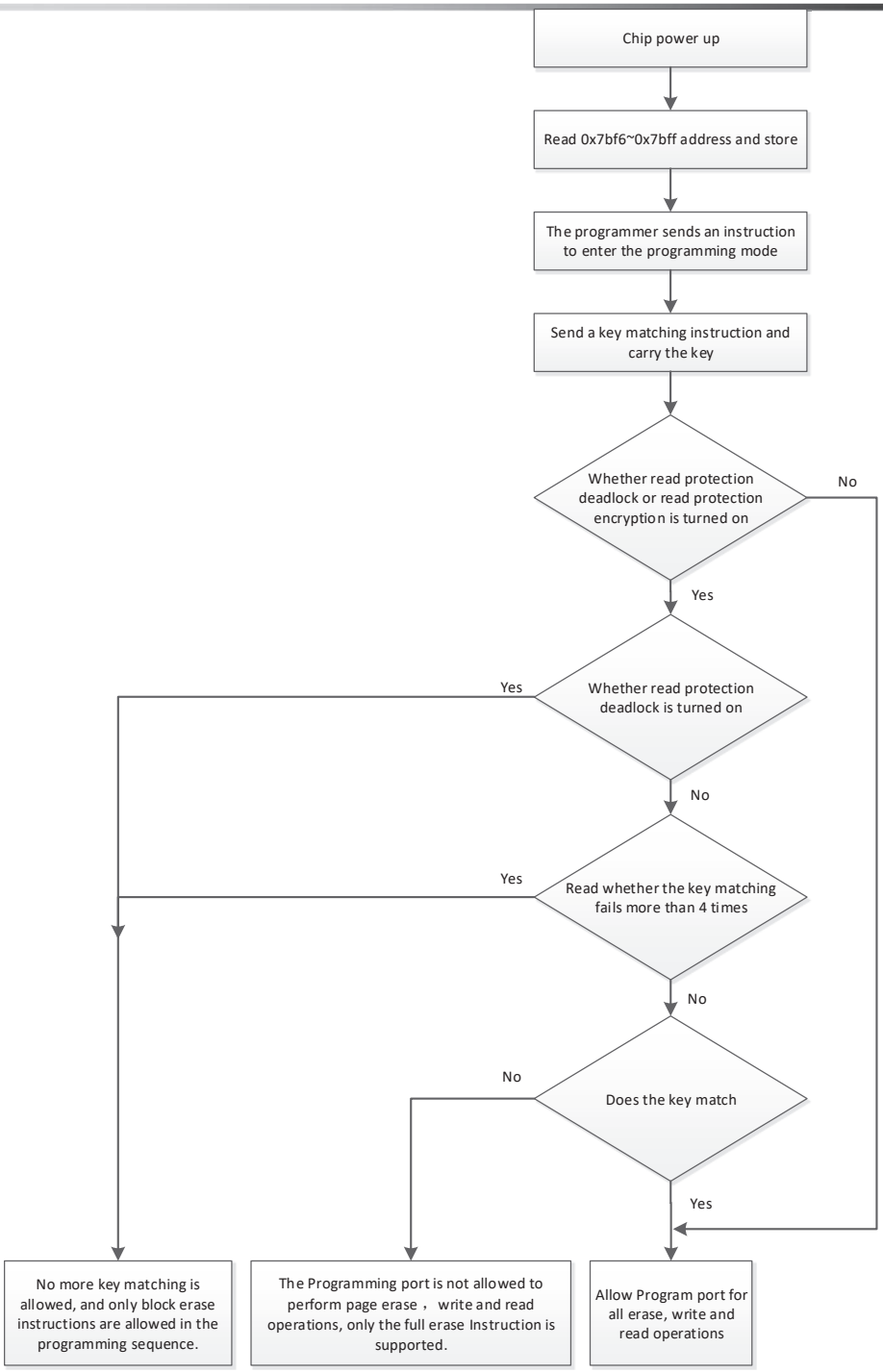


Figure 0.4 Read Protection Control Flow Chart in Programming Mode

Note

- When the Read Protection is valid, the debug port is invalid and can not work upon any commands.
- When the Read Protection is invalid, all space can be read by CPU, SWE,PG except the address from 0x7BE0 to 0x7BFF which can not be read by CPU.

- When the Read Protection is valid, CPU cannot read the address from 0x7BE0 to 0x7BFF. SWE can not be read in any way except after keys match in programming mode (Read protection lock dose not support KEY Match to unlock).
- No limit on EEPROM (0x7c00 to 0x7fff) and INFORMATION (0x8000 to 0x81ff) to be read at any time.
- If the Read Protection Lock is enabled, it is not allowed to match keys in code mode and support no operations except the command Erase all.
- If only the Read protection encryption is enabled, the code mode allows erase page, write page and read page when keys match, otherwise just Erase all is allowed.
- If only the Read protection encryption is enabled, code mode allows keys match operation. If the number of mismatches is greater than 4, the matching operation is no longer allowed, and only the full erase instruction is allowed to execute.
- The configuration bit of Read protection lock is 0x68. The configuration bit of read protection encryption is 0x83.
- Read the storage value of protection and key once after power-on, change key or enable read protection in the programming mode, and power-on again to make it valid.
- Write key at the programming interface,write hex file through programming at a time instead of writing by Trim Register.

3.3 SRAM

SRAM functions as a memory. As the internal RAM and the extension RAM of BS9000AM series, the main features of SRAM are as follows:

- System clock: 24MHz.
- Including internal RAM (iram) and extension RAM (xram).
  - The size of Internal RAM is 256 Bytes, 8-bit address ranging from 00~FFH.
  - The size of extension RAM is 2048 Bytes, 11-bit address ranging from 000~ 7FFH.
- Part of the system register XRAM_SFR uses XRAM bus, with address ranging from 2000~FFFFH.

Reserve stack space during programming to avoid the stack overflow. The first address of the stack is distributed automatically according to program when the C programming language is used, but it must store at DATA sector or IDATA sector. In Keil, set up the first address of stack in STARTUP.A51.

The SRAM address space is divided as follows:

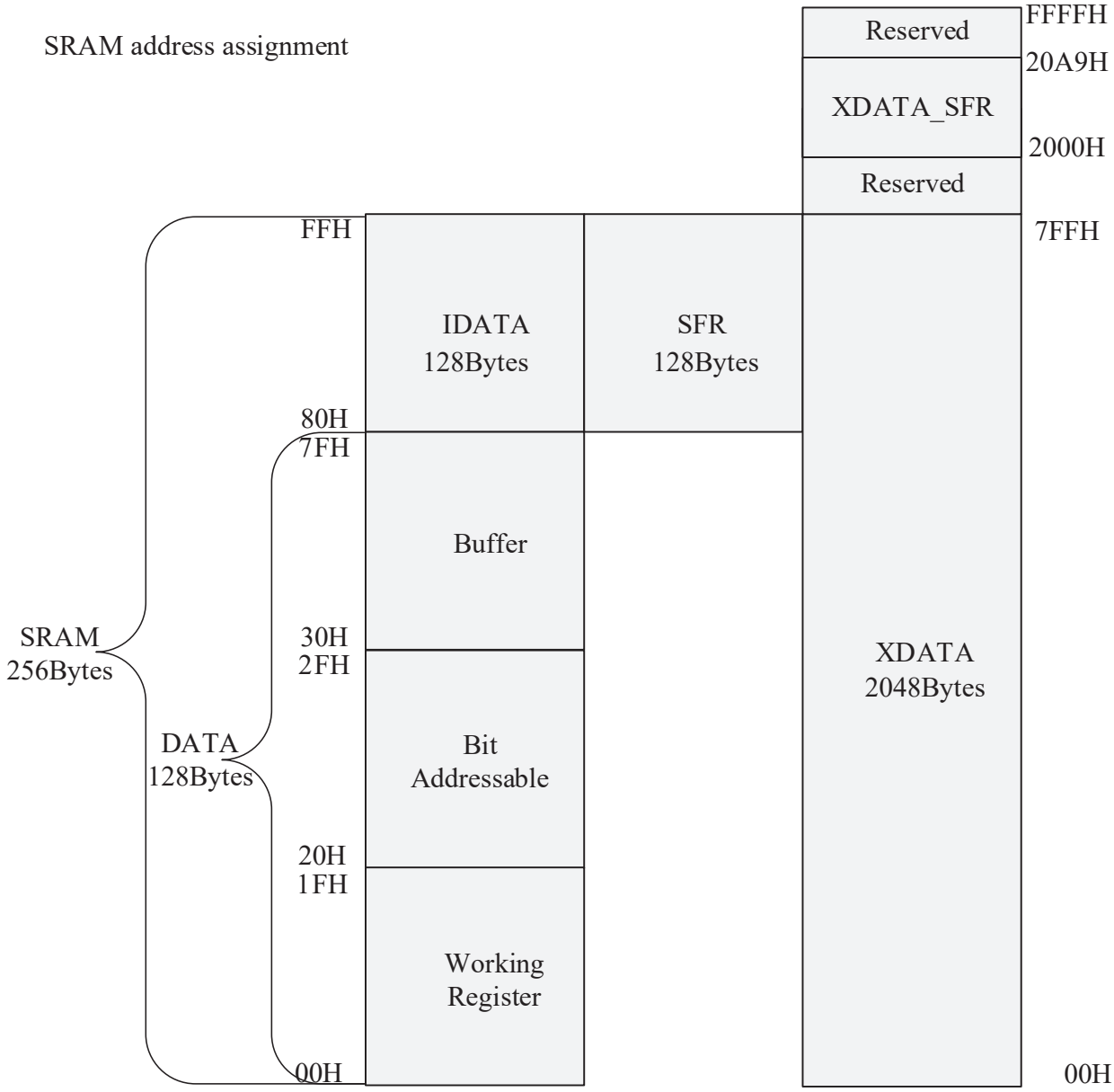


Figure 0.5 SRAM Address Allocation

The total size of internal SRAM is 256 Bytes with the address from 00H to FFH, including operation register group, bit address and buffers (stacks included).

- Internal Low 128 bytes: from 00H to 7FH totaling128 Bytes. Read out and write data through immediate addressing or indirect addressing.
- Internal High 128 bytes: from 80H to FFH totaling 128 Bytes. Read out and write data through operation register indirect addressing.

SFR: Special functional register at address of 80H to FFH, using direct addressing to read and write data.

XDATA has a total size of 2048Bytes at address from 0000H to 07FFH. User can totally use this block. It uses data pointer or register addressing to read and write data.

XRAM_SFR is at address from 2000H to FFFFH. User can totally use this block. It uses data pointer or register addressing to read and write data.

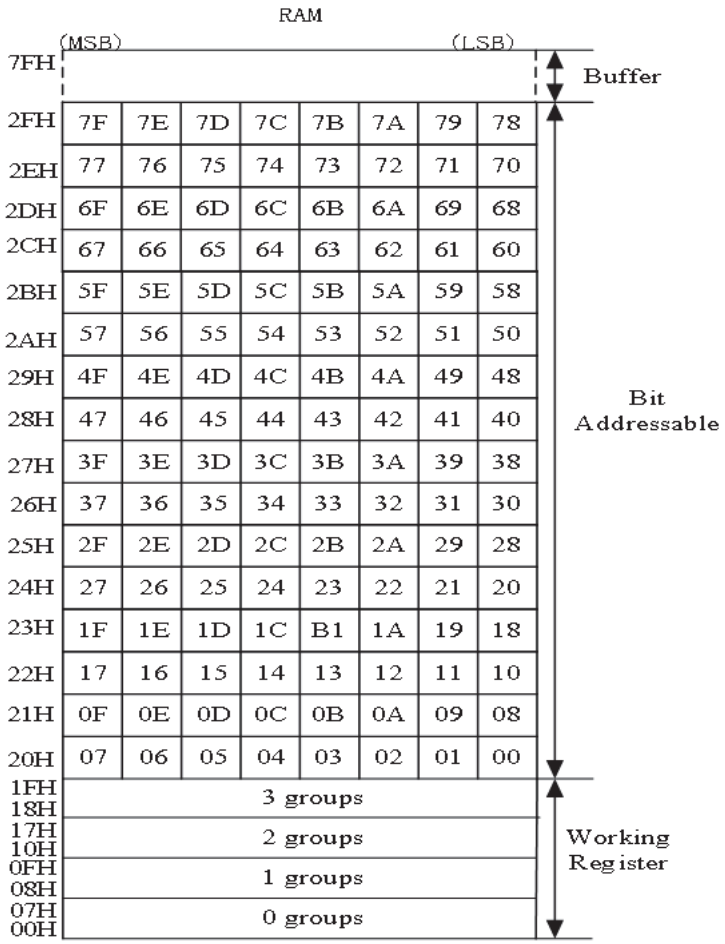


Figure 0.6 Bit Addressable Distribution

The table below gives addressing methods of 3 modules from RAM:

Region	Addressing Method

DATA	MOV A,direct
	MOV direct,A
	MOV direct,#data
	MOV direct1,direct2
	MOV Rn,direct
	MOV direct,Rn

IDATA	MOV A,@Ri
	MOV @Ri,A
	MOV direct,@Ri
	MOV @Ri,direct
	MOV @Ri,#data

XDATA	MOVX @DPTR,A

	MOVX A,@DPTR

Table 0.7 RAM Evaluation Instructions

3.4 EEPROM

EEPROM is the last 1KBytes of the Main Block.

EEPROM Program: User code can conduct page erase and byte write operations in EEPROM space.

Please use the program to directly access and read out the value of EEPROM.

The table below shows 2 BLOCK positions from EEPROM:

EEPROM	BLOCK Address	Description
BLOCK0	0x7E00	Every block has a size of 512Bytes. One erase operation can erase a space of 512Bytes. SPROG_ADDR_H[1]=1, select block1. SPROG_ADDR_H[1]=0, select block0. SPROG_ADDR_H[0] and SPROG_ADDR_L[7:0] select page address SPROG_ADDR_H[7:2] is invalid.
BLOCK1	0x7C00	

Table 0.8 EEPROM Address

Notes:

1. Every address in the EEPROM can only be written into once after the erase operation, erase it first if there is need to write again.
2. For a EEPROM, data can be written into in batches after the erase operation.

3.4.1 Page Erase

1. SPROG_TIM[4:0] = 0 to 20 (recommended setting as 10: 30ms). The byte write time is fixed to 62μs. SPROG_TIM is configured only once in the initiation of the main().
2. Disable the general interrupt
3. Configure SPROG_ADDR_L=0x00.
4. Configure SPROG_ADDR_H=0x0x(bit[1];0:select block0;1:select block1), choose the page needs to be erased.
5. Configure SPROG_LIN_CMD=0x96, EEPROM page erase command.
6. Write into 4 consecutive NOP commands.
7. When the erase starts, CPU enters into IDLE mode and exits automatically after it finished.
8. For further data erasing, jump to step 2.

9. Configure SPROG_ADDR_H=0x00,SPROG_ADDR_L=0x00, recover the interrupt.
10. Enable the general interrupt.

3.4.2 Byte Programming

1. SPROG_TIM[4:0] = 0 to 20 (Recommended setting as 10: 30ms). The Byte write time is fixed to 62μs. SPROG_TIM is configured once in the main() initiation. Disable interrupt.
2. Disable the general interrupt.
3. Configure SPROG_ADDR_L, SPROG_ADDR_H, select the address of the byte programming.
4. Configure SPROG_DATA, EEPROM programs the data to be written into.
5. Configure SPROG_LIN_CMD=0x69, EEPROM bit programming command.
6. Write into 4 consecutive NOP commands.
7. When the write starts to program, CPU enters into IDLE mode and exits automatically after it completes.
8. For further data programming, jump to step 2.
9. Configure SPROG_ADDR_H=0x00, SPROG_ADDR_L=0x00, recover the interrupt.
10. Enable the general interrupt.

3.5 IAP Operation

FLASH IAP routines as follows:

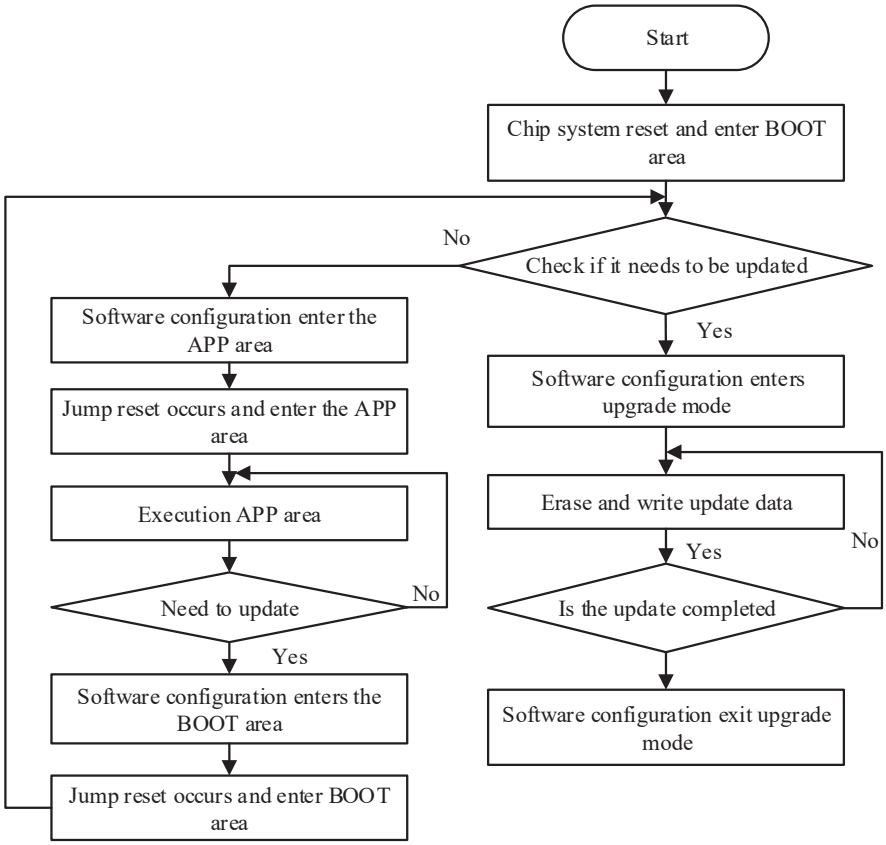


Figure 0.9 FLASH IAP Operation Flow Chart

3.5.1 Program Upgrade

At the address block that write protection is invalidated, page erase and byte write operations are allowed through the program to upgrade the BOOT program.

- Enter BOOT upgrade steps:
  Write 0x12, 0x34, 0x56, 0x78 and 0x9A into SPROG_LIN_CMD consecutively, enter BOOT upgrade mode of the FLASH, when all space is write-protected, it can not enter BOOT upgrade mode.
- Exit BOOT upgrade steps:
  Write 0xFE, 0xDC, 0xBA, 0x98 and 0x76 into SPROG_LIN_CMD consecutively, exit FLASH BOOT upgrade mode.
- The steps of Erase and byte programming is the same as the EEPROM.
- {SPROG_ADDR_H[6:0], SPROG_ADDR_L[7:0]} multiplexed as the FLASH space Address at 0x0000~0x79FF.
- Erase one page once, the size of 1 page is 512Bytes.

3.5.2 FLASH IAP Erase Operation

FLASH_BOOT Upgrade Steps:

1. SPROG_TIM[4:0] = 0~20 (recommended setting 30ms). The byte writing time is fixed to 62μs. SPROG_TIM is configured only once in the initiation of the function main().
2. Disable the interrupt.
3. Configure SPROG_ADDR_L = 0x00.
4. Configure SPROG_ADDR_H[5:1], select erase this page.
5. Configure SPROG_LIN_CMD = 0x96, FLASH page erase command.
6. Write 4 consecutive NOP commands.
7. When the erase operation starts, CPU disables F_sys_clk and enables it once finished.
8. For further data erasing, jump to step 2.
9. Configure SPROG_ADDR_H=0x00, SPROG_ADDR_L=0x00, recover the interrupt.
10. Enable General Interrupt.

3.5.3 FLASH IAP bit Write

In FLASH_BOOT Upgrade mode:

1. SPROG_TIM[4:0] =0 to 20 (recommended 30ms). The byte writing time is fixed to 62μs. SPROG_TIM is configured only once in the initiation of the function main().
2. Disable General Interrupt.

3. Configure SPROG_ADDR_L, SPROG_ADDR_H, selects the address of the programming.
4. Configure SPROG_DATA, FLASH programs the data to be written into.
5. Configure SPROG_LIN_CMD = 0x69, FLASH bits programming command.
6. Write into 4 NOP instructions.
7. Start to write, CPU disable F_sys_clk and open it again once completed.
8. If continue to write data, jump to step 2.
9. Configure SPROG_ADDR_H=0x00, SPROG_ADDR_L=0x00, recover the interrupt.
10. Enable General Interrupt.

3.5.4 FLASH IAP Operation

FLASH IAP operations are the following:

Instruction	Response	Data
Enter Upgrade Mode	FLASH_BOOT_EN = 1	0x12、0x34、0x56、0x78、0x9A
Exit Upgrade Mode	FLASH_BOOT_EN = 0	0xFE、0xDC、0xBA、0x98、0x76

Table 0.10 FLASH IAP Operations

Instruction descriptions

- Enter upgrade mode: SPROG_LIN_CMD write sequence: 0x12, 0x34, 0x56, 0x78, 0x9A.
- Exit upgrade mode: SPROG_LIN_CMD write sequence: 0xFE, 0xDC, 0xBA, 0x98, 0x76.

Response

- FLASH_BOOT_EN = 1 indicates entering the FLASH BOOT upgrade mode;
- FLASH_BOOT_EN = 0 indicates exiting the FLASH BOOT upgrade.

Note:

- It is necessary to write the instruction data of SPROG_LIN_CMD and BOOT_LIN_CMD in order, otherwise it requires to rewrite.
- MCU operating voltage: 3.0V to 5.5V. MCU may occur error at 1.5V~3.0V, resulting in wrong upgrade. It is recommended to test the voltage of ADC or LVDT before the IAP operation, if it is lower than 3.0V, do not conduct the IAP.
- It is recommended to mask the interrupt during the upgrading to ensure that the IAP operation will not be interrupted, recover the interrupt to once finished. Check data after the upgrading to make sure it is correct.

3.6 Registers for Memory

3.6.1 Interrupt Vector Table Offset Register (OFFSET_ADDR_L)

XRAM_SFR address: 0x2020

Bit No.	7	6	5	4	3	2	1	0
Symbol	OFFSET_ADDR[7:0]							
Read/Write	Read/write							
Power-on initial value	0							

No.	Symbol	Description
7~0	OFFSET_ADDR[7:0]	Vector table offset interrupt address (bit 7-0)

3.6.2 Vector Table Offset Interrupt Address Register (OFFSET_ADDR_H)

XRAM_SFR address: 0x2021

Bit No.	7	6	5	4	3	2	1	0
Symbol	OFFSET_ADDR[15:8]							
Read/Write	Read/Write							
Power-on initial value	0							

No.	Symbol	Description
7~0	OFFSET_ADDR[15:8]	Vector table offset interrupt address high 8 bits

3.6.3 Address Control Register (SPROG_ADDR_H)

XRAM_SFR address: 0x2062

Bit No.	7	6	5	4	3	2	1	0
Symbol	-						-	-
Read/Write	Read/Write						Read/Write	Read/Write
Power-on initial value	0						0	0

Bit No.	Symbol	Description
7~2	--	--
1	--	Selects EEPROM block (for page erase and bit program) 0: select block0 1: select block1
0	--	Indicates EEPROM block address SPROG_ADDR[8] I n FLASH BOOT upgrade mode, {SPROG_ADDR_H[6:0], SPROG_ADDR_L} is multiplexed with all addresses of FLASH ranging from 0x0000 to 0x79FF

3.6.4 Address Control Register (SPROG_ADDR_L)

XRAM_SFR address: 0x2063

No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Power-on initial value	0							

Bit No.	Symbol	Description
7~0	--	Indicates EEPROM block address bits 7 to 0 {SPROG_ADDR_H[0], SPROG_ADDR_L[7:0]}indicates block address

3.6.5 Data Register (SPROG_DATA)

XRAM_SFR address: 0x2064

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Power-on initial value	0							

Bit No.	Symbol	Description
7~0	--	EEPROM programming: data to be written

3.6.6 Command register (SPROG_CMD)

XRAM_SFR address: 0x2065

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Power-on initial value	0							

Bit No.	Symbol	Description
7~0	--	Write into 0x96: EEPROM page erase Write into 0x69: EEPROM bit programming Write 0x12, 0x34, 0x56, 0x78, 0x9A in order to enter into FLASH BOOT Upgrade Mode. Write 0xFE, 0xDC, 0xBA, 0x98, 0x76 in order to exit FLASH BOOT Upgrade Mode. When all space is write-protected, BOOT upgrade mode can not be entered.

3.6.7 Erase and Write Time Control Register (SPROG_TIM)

XRAM_SFR address: 0x2066

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	-
Read/Write	-	-	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Power-on initial value	-	-	-	0	1	0	1	0

Bit No.	Symbol	Description
7~5	--	Reserved
4~0	--	(prog set as 30us) bit[4:0]: 0~20: Erase time = 20 to 40ms (in increments of 1ms) >20: Erase time = 30ms

3.6.8 BOOT Status Register (FLASH_BOOT_EN)

XRAM_SFR address: 0x2067

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	FLASH_BOOT_EN
Read/Write	-	-	-	-	-	-	-	Read
Power-on initial value	-	-	-	-	-	-	-	0

Bit No.	Symbol	Description
0	FLASH_BOOT_EN	1: Enter FLASH BOOT Upgrade mode. 0: Exit FLASH BOOT Upgrade mode. In this mode, registers SPROG_ADDR_H, SPROG_ADDR_L, SPROG_DATA, SPROG_LIN_CMD, SPROG_TIM multiplexed as BOOT upgrade {SPROG_ADDR_H, SPROG_ADDR_L} multiplexed with address space range 0x0000~0x79FF in FLASH.

3.6.9 Write Protect Range Configuration Register (FLASH_WPROTECT_PAGE)

XRAM_SFR address 1: 0x7A00

XRAM_SFR address 2: 0x7A01

Note: The data at address 2 is the inverse of the data in address 1, that is, the data at address 2 is complementary to the data at address 1. The configuration is valid only when both address 1 and address 2 are configured.

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	CFG_WPROTECT_PAGE					
Read/Write	-	-	Read/Write					
Power-on initial value	-	-	1	1	1	1	1	1

Bit No.	Symbol	Description
5~0	CFG_WPROTECT_PAGE	Flash address write protection range configuration register by page.Full-space protection is enabled by default.

4 System

4.1 Clock

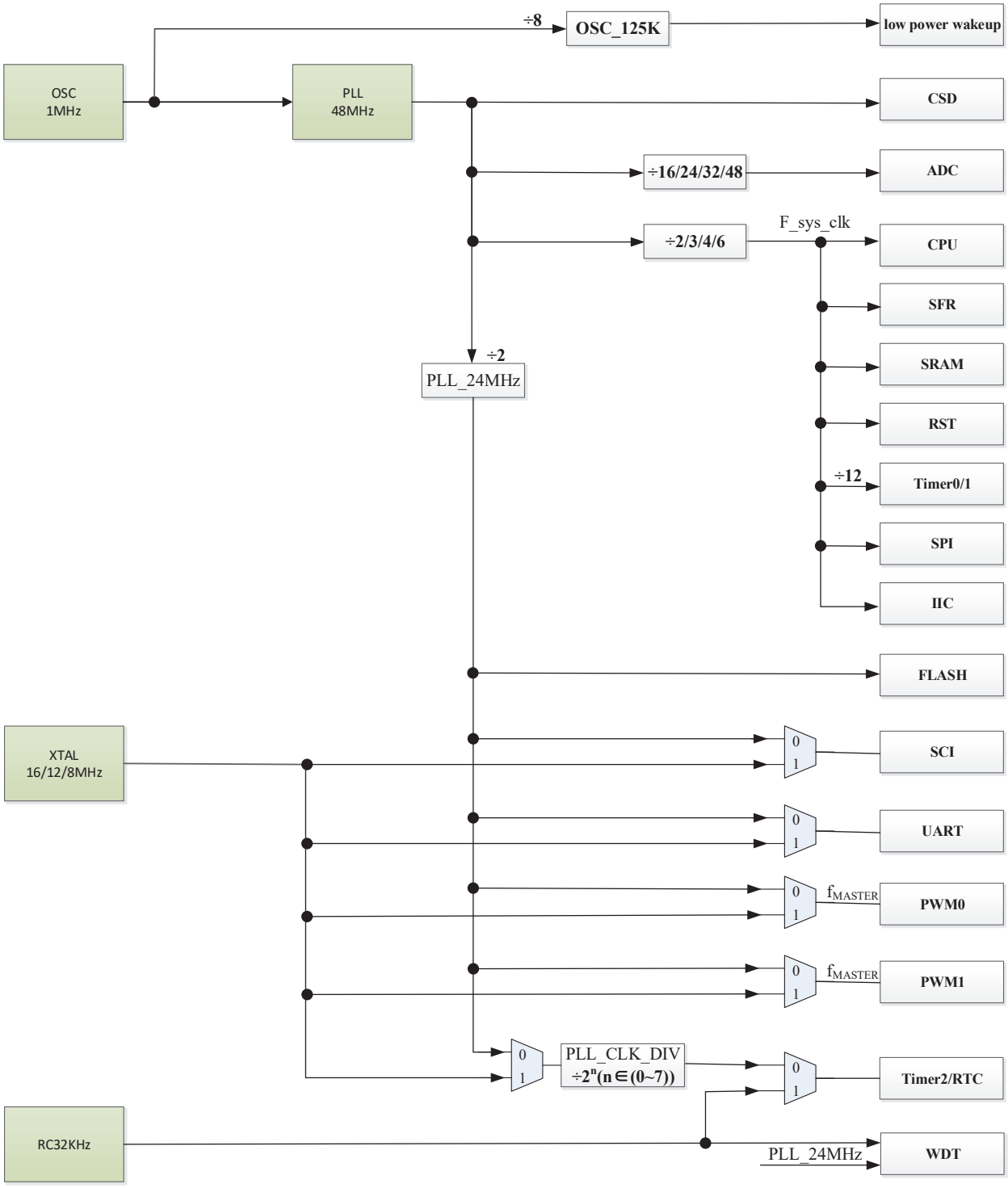


Figure 0.1 Clock Block Diagram

OSC_125K: Low Power Consumption Wake-up Clock.
F_sys_clk: System Clock, select frequency from 24MHz/16MHz/12MHz/8MHz.
PLL_48MHz: PLL generated 48MHz clock used as CSD and system clock after frequency division.
PLL_24MHz: PLL generated 24MHz clock directly used as controllers like UART, FLASH and PWM.
XTAL16MHz/12MHz/8MHz: mainly used for peripheral clock like Timer2, SCI and PWM.
PLL_CLK_DIV: divide the frequency of PLL_24Mhz and XTAL,where n is from 0 to 7 for 2^n
RC32KHz: The CLOCK frequency is 32768Hz, used as clock source of WDT, Timer2 and RTC.

Function	Clock Source
Low power consumption Wake-up	OSC_125K
CSD	PLL_48MHz
ADC	3MHz/2MHz/1.5MHz/1MHz
CPU	System Clock F_sys_clk(24MHz/16MHz/12MHz/8MHz)
SFR	System Clock F_sys_clk(24MHz/16MHz/12MHz/8MHz)
SRAM	System Clock F_sys_clk(24MHz/16MHz/12MHz/8MHz)
RST	System Clock F_sys_clk(24MHz/16MHz/12MHz/8MHz)
Timer0/1	System Clock F_sys_clk(24MHz/16MHz/12MHz/8MHz)/12
SPI	System Clock F_sys_clk(24MHz/16MHz/12MHz/8MHz)
IIC	System Clock F_sys_clk(24MHz/16MHz/12MHz/8MHz)
FLASH	PLL_24MHz
SCI	PLL_24MHz or XTAL(16MHz/12MHz/8MHz)
UART	PLL_24MHz or XTAL(16MHz/12MHz/8MHz)
PWM0	PLL_24MHz or XTAL(16MHz/12MHz/8MHz)
PWM1	PLL_24MHz or XTAL(16MHz/12MHz/8MHz)
TIMER2/RTC	PLL_CLK_DIV Frequency division of PLL_24MHz/XTAL or RC32KHz
WDT	PLL_24MHz or RC32KHz

Table 0.2 List of Peripheral Clock Source

4.2 Reset System

The digital pasrt of BS9000AMxx has 7 reset sources: Power-on Reset (po_n), Brown-out Reset (bo_n), Program Reset (prog_en), Software Reset (soft_rst), WDT Reset (wdt_rst), Address Overflow Reset (addr_overflow) and Debug Reset (debug_rst_out_n). If any kind of reset occurs and generates the global reset signal, it will reset the device. According to reset status register, check which reset has happened. Reset status bits require software to clear. See Reset diagram below.

See POR/BOR sequence diagram below.

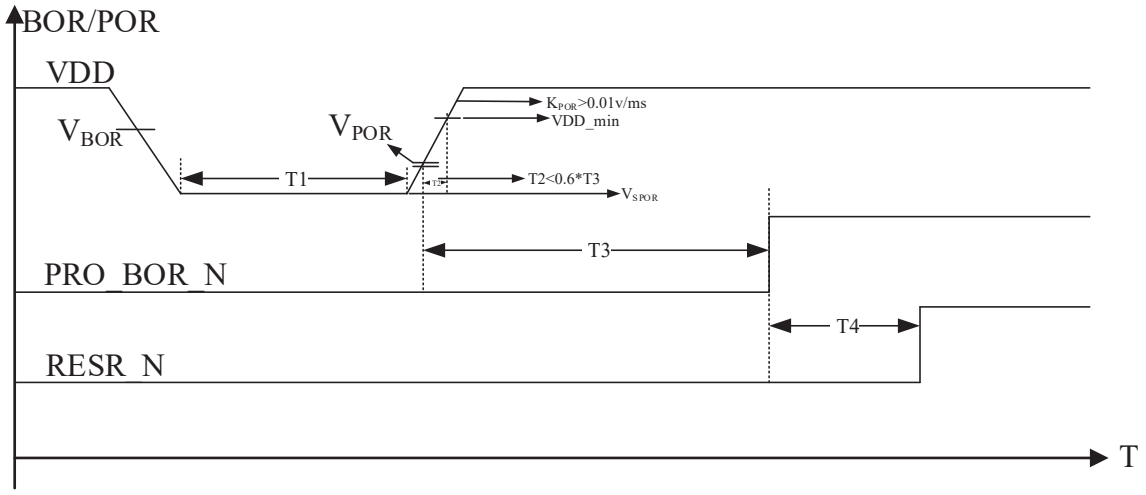


Figure 0.3 POR Diagram

See PON/BOR parameters below.

Symbol	Parameter	Min	Typ	Max	Unit
VSPOR	POR Startup voltage	-	-	300	mV
KPRO	POR slew rate	0.38	-	-	V/ms
VPOR	POR Voltage	1.21	1.59	2.04	V
VBOR	BOR Voltage (±10%), Hysteresis is 0.2V	-	2.6	-	V
	BOR Voltage (±10%), Hysteresis is 0.2V	-	2.8	-	V
	BOR Voltage (±10%), Hysteresis is 0.2V	-	3.7	-	V
	BOR Voltage (±10%), Hysteresis is 0.2V	-	4.2	-	V
VDD_min	Minimum Operating Voltage	3.0	-	-	V
T1	VDD hold time for VSPOR	0.1	-	-	ms
T2	VPOR to VDD_min time	-	-	0.6*T3	ms
T3	POR_BOR_N hold time	-	7.8	-	ms
T4	General reset valid time	-	3	-	ms

Figure 0.4 POR Characteristics Parameters

When VDD is impacted by loads or severe disturbance, if voltage falls into the voltage deadtime and the device is not within the operating voltage range, it may cause system fault, such as EEPROM data loss. The BOR serves as a monitor when VDD falls to BOR voltage, MCU can trigger BOR in advance in case the system faults occur.

Tips on preventing the voltage falling into deadtime which may cause the system fault: ramp up the falling slope.

4.3 LVDT

BS9000AMxx supports Low Voltage Detection, which monitors the voltage status change. It has 5 different voltage levels: 3.0V/3.3V/3.6V/3.9V/4.5V (Preset Point voltage drop interrupt, 0.1V hysteresis generates the corresponding voltage rising interrupt). When the voltage monitor configures the threshold value above-mentioned, voltage drops to this threshold will trigger Low Voltage interrupt. During the low voltage interrupt, the system can handle it according to application requirements.

See LVDT configuration flow chart as bellow.

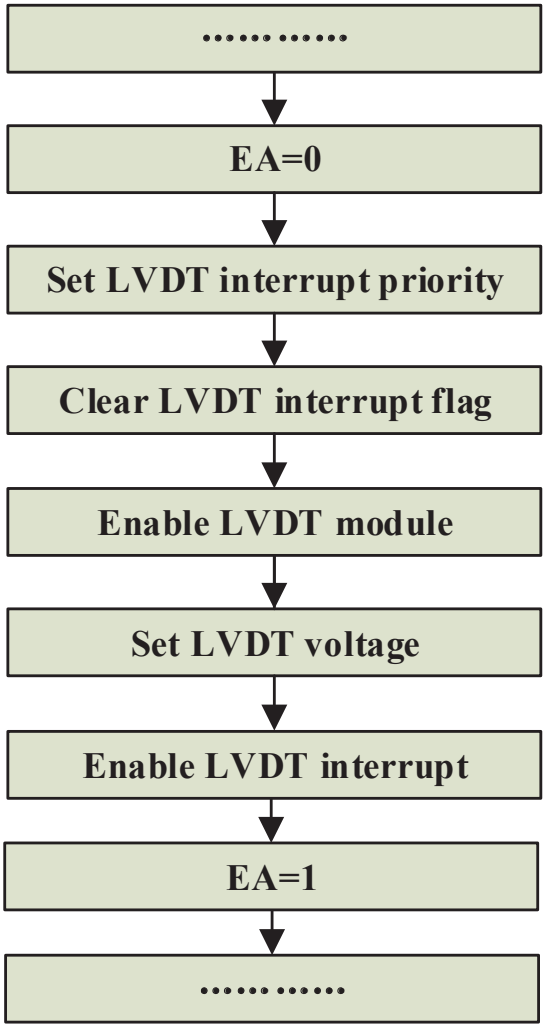


Figure 0.5 LVDT Configuration

4.4 Operating Modes

BS9000AMxx supports 3 operating modes based on different situations.
BS9000AMxx supports register SYS_CLK_CFG, configure its Bit 2 to enter Wait Mode.
BS9000AMxx supports PCON register, configure its Bit 0 and MCU enters sleep mode.

- **Active Mode:** Modules operates normally, their functions are controlled by software configurations.
- **Wait Mode:** Write 1 into SYS_CLK_CFG .2 to enter wait mode. In wait mode, analog clock is not OFF. To reach the low consumption state, only disable the clocks related to digital system core. Except the Core-related modules and SPI module, the rest modules can work and exit the wait mode by interrupt.
- **Sleep Mode:** Configure IDLE_EN=1 to enter sleep mode. Then, disable OSC1M and PLL, enable RC32K, enable WDT/TIMER2/RTC, IIC interrupt/SCI edge interrupt set to wake up, disable CPU and the rest digital modules.

Both Reset and valid interrupt can exit wait or sleep mode. The wake-up interrupt differs in different modes. See below:

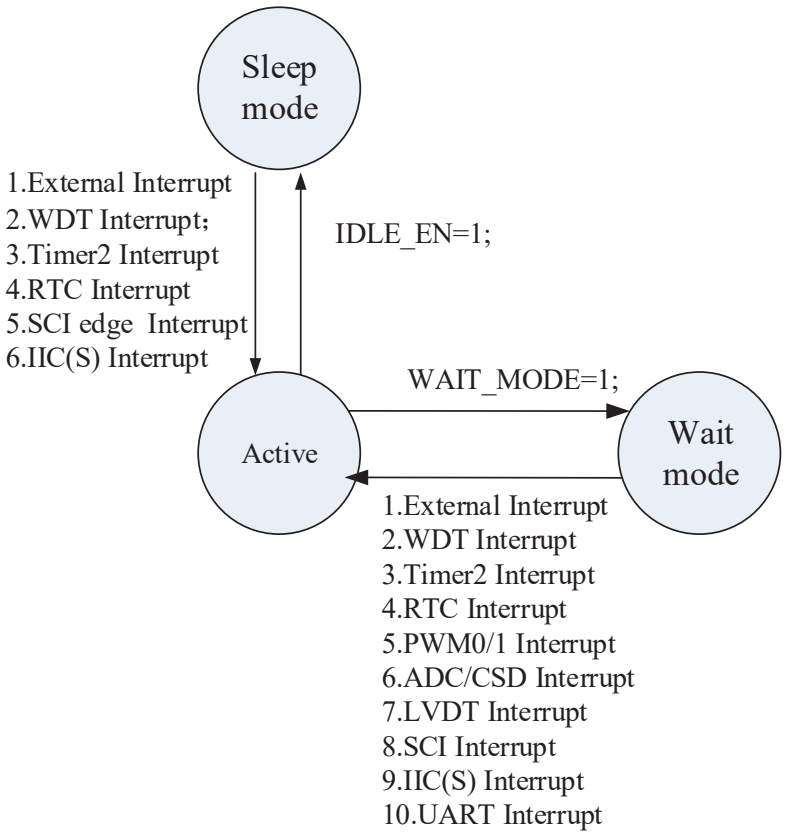


Figure 0.6 Operating Mode Conversion

Ways to exit the Waiting Mode

- Enable IIC, External Interrupt0, External Interrupt1, External Interrupt2, External Interrupt3,

WDT, Timer2, RTC, CSD, ADC, LVDT, UART, SCI and PWM0/1. Any kind of interrupt among them can wake up the device to exit the wait mode. CPU executes the interrupt program.

Ways to exit the Sleep Mode

- Enable External Interrupt 0, External Interrupt1, External Interrupt2, External Interrupt3, IIC, WDT, Timer2, RTC and SCI edge. Any kind of interrupt among them can wake up the device to exit the sleep mode. When interrupt occurs, CPU executes interrupt vector-related interrupt service routine. After executing RETI return command, CPU comes back to run next program, which is after the program allowing CPU to enter sleep mode.

Table 4.9 shows the clock source in different operating states.

Modes	conditions	Results	
Active	IDLE_EN=0; WAIT_MODE=0	RC32K	Enable
		XTAL	Depend on software configuration
		OSC1M	Enable
		PLL	Enable
Wait	IDLE_EN=0; WAIT_MODE=1	RC32K	Enable
		XTAL	Depend on software configuration
		OSC1M	Enable
		PLL	Enable
Sleep	IDLE_EN=1	RC32K	Enable
		XTAL	Depend on software configuration
		OSC1M	Disable
		PLL	Disable

Table 0.7 Clock Source in Different Modes

Table 4.8 shows the states of digital modules in different modes.

NO	Module	Clock source	Operating state		
			Active	Wait	Sleep
1	s8051	F_sys_clk	√	×	×
2	Timer0	F_sys_clk/12	According to program configuration	×	×
3	Timer1	F_sys_clk/12	According to program configuration	×	×

4	Timer2/RTC	RC32KHz, XTAL, PLL_24MHz	According to program configuration	According to program configuration	According to program configuration
5	WDT	24MHz, RC32K	According to program configuration	According to program configuration	According to program configuration
6	PWM0/1	F_sys_clk, XTAL	According to program configuration	According to program configuration	×
7	UART	F_sys_clk, XTAL	According to program configuration	According to program configuration	×
8	SCI	F_sys_clk, XTAL	According to program configuration	According to program configuration	According to program configuration
9	SPI	F_sys_clk	According to program configuration	According to program configuration	×
10	IIC(S)	F_sys_clk	According to program configuration	According to program configuration	According to program configuration
11	ADC	3MHz/2MHz/1.5MHz/1MHz	According to program configuration	According to program configuration	×
12	CSD	PLL_48MHz	According to program configuration	According to program configuration	×

Table 0.8 Digital Modules in different modes

F_sys_clk: System clock with frequency 24MHz/16MHz/12MHz/8MHz to select.
XTAL: external crystal with frequency 6MHz/12MHz/8MHz to select.

4.5 Watchdog Timer (WDT)

WDT circuit uses RC32K or PLL_24MHz to set time as 2^n*18ms (n=0, 1, 2, 3, 4, 5, 6, 7) where n is the set value of timer configuration register.
WDT reset signal consists of general reset and configuration reset. The signal supports synchronous release by WDT clock in the Reset module. Clear operation is generated each time CPU configures WDT configuration register, then WDT restarts. Concurrently, WDT counter has its counter enable control. When the counter is enabled and WDT generates overflow (reset or interrupt), the WDT counter restarts as long as the counter is not disabled.

WDT overflow signal can be categorized into two types due to the speciality of system application.

- When system is under normal mode and WDT occurs overflow, then the overflow signal is the WDT overflow reset signal. Since the WDT overflow reset impacts the general reset, at this time the system implements general reset and reload the configuration information.
- If WDT overflows when system is in sleep or wait mode, then the overflow signal is the WDT interrupt signal, which wakes up the device to exit sleep or wait mode and executes WDT interrupt services function.

4.6 System Registers

4.6.1 Software Reset Register (SOFT_RST)

XRAM_SFR address: 0x2022

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Power-on initial value	0							

Bit No.	Symbol	Description
7~0	--	Software reset register will trigger software reset if the register value is 0x55.

4.6.2 Reset Flag Register (RST_STAT)

XRAM_SFR address: 0x2023

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	DEBUG_F	SOFT_F	PROG_F	ADDROF_F	BO_F	PO_F	WDTRST_F
Read/Write	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Power-on initial value	-	0	0	0	0	0	1	0

Bit No.	Symbol	Descriptions
6	DEBUG_F	1: enables debug reset 0: stay in the previous state
5	SOFT_F	1: enables software reset 0: stay in the previous state
4	PROG_F	1: enables program reset 0: stay in the previous state
3	ADDROF_F	1: enables PC overflow reset 0: stay in the previous state
2	BO_F	1: enables Brown-out reset 0: stay in the previous state
1	PO_F	1: enables Power-on reset 0: stay in the previous state
0	WDTRST_F	1: enables WDT overflow reset 0: stay in the previous state

Note:
Reset value represents the reset values in different modes. For example, the value of rst_state is 0x02 for Power-on Reset; for other reset modes, the corresponding reset flag bit of rst_state is 1, other flag indicates it stays in the previous mode.

4.6.3 Module Switching Control Register (PD_ANA)

XRAM_SFR Address: 0x2024

Bit No.	7	6	5	4
Symbol	-	-	LDO_LOAD2	LDO_LOAD1
Read /write	-	-	Read /write	Read /write
PO initial value	-	-	0	0
Bit No.	3	2	1	0
symbol	XTAL_SEL	PD_XTAL	PD_CSD	PD_ADC
Read /write	Read /write	Read /write	Read /write	Read /write
PO initial value	0	1	1	1

Bit No.	Symbol	Description
5	LDO_LOAD2	In WAIT mode, DC Loads 450μA control, reset value is 0. 0: indicates DC loads 450μA Control 1: No DC loads
4	LDO_LOAD1	In Active mode, DC Loads 450μA control, reset value is 0.

		0: indicates DC loads 450μA 1: No DC loads
3	XTAL_SEL	Oscillator/Crystal selects register 0: selects crystal 1: selects oscillator
2	PD_XTAL	Crystal circuit control register 1: Disabled; 0: enabled (disabled by default)
1	PD_CSD	Analog CSD (control system designed) register PD_CSD=0 CSD in active mode PD_CSD=1 CSD does not work
0	PD_ADC	Analog ADC control register PD_ADC=0 ADC in active mode PD_ADC=1 ADC does not work

4.6.4 BOR Threshold Selection Register (BOR_SEL)

XRAM_SFR Address: 0x2025

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	SEL_BOR_VTH	
Read/Write	-	-	-	-	-	-	Read/Write	Read/Write
Po initial value	-	-	-	-	-	-	0	0

Bit No.	Symbol	Description
1~0	SEL_BOR_VTH	BOR threshold selection 00: 2.6V; 01:2.8V; 10: 3.7V; 11: 4.2V;

See the BOR voltage threshold selection table below.

Threshold selection SEL_BOR_VTH	BOR			
	BOR threshold (V)	Reset threshold(V)	voltage hysteresis(mV)	Delay (μs)
00	2.6	2.7	114.7	72.62
01	2.8	2.9	132	81.78
10	3.7	3.8	155.4	125.8
11	4.2	4.3	101.5	148.6

Table 0.11 BOR Threshold Selection Table

4.6.5 LVDT Control Register (LVDT_SEL)

XRAM_SFR Address: 0x2026

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	PD_LVDT	SEL_LVDT_VTH		
Read/Write	-	-	-	-	Read/Write	Read/Write	Read/Write	Read/Write
Po initial value	-	-	-	-	1	0	0	0

Bit No.	Symbol	Description
3	PD_LVDT	LVDT control register 1: disabled; 0: enabled (disabled by default)
2~0	SEL_LVDT_VTH	LVDT threshold selection 000：3.0V；001：3.3V；010：3.6V； 011：3.9V；1xx：4.5V；

See Power-on threshold value selection table below.

Threshold Selection SEL_LVDT_VTH	LVDT			
	BOR threshold (V)	Reset threshold (V)	Hysteresis (mV)	Delay (μs)
000	3.0	3.1	98.92	6.622
001	3.3	3.4	120.5	7.639
010	3.6	3.7	105.9	8.536
011	3.9	4.0	84.09	9.594
1xx	4.5	4.6	108.9	11.32

Table 0.12 Po threshold selection table

4.6.6 Test Clock Register (SFR_TEST_SEL1)

XRAM_SFR address: 0x201E

Bit No.	7	6	5	4
Symbol	-	-	TEST_XTAL_I	TEST_XTAL_EN
Read/Write	-	-	Read/Write	Read/Write
Po initial value	-	-	0	0
Bit No.	3	2	1	0
Symbol	TEST_CLK_EN	TEST_CLK_SEL		
Read/Write	Read/Write	Read/Write		

Po initial value	0	0

Bit No.	Symbol	Description
5	TEST_XTAL_I	Oscillator bias current test. 0: No test. 1: Test.
4	TEST_XTAL_EN	Whether enable XTAL when selecting test clock output enable, the default value is 0. 1: enable 0: disable Note: it is only valid when outputting the test clock (TEST_CLK_EN=1)
3	TEST_CLK_EN	Select if output test clock. 1: output enable 0: output disable. The default value is 0.
2~0	TEST_CLK_SEL	PA2/PA3 test at the same time. 000: selects 1MHz oscillator clock (OSC_1M); 001: selects 32KHz oscillator clock (OSC_32k); 010: selects 2MHz clock (PLL_24M after frequency is divided by 12)011: selects 1MHz oscillator clock (OSC_1M) 100: selects XTAL clock (XTAL frequency is divided by 32)

4.6.7 Sleep Control Register (PCON)

Address: 0x87

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	SLEEP_MODE
Read/Write	-	-	-	-	-	-	-	Read/Write
Po initial value	-	-	-	-	-	-	-	0

Bit No.	Symbol	Description
7~1	--	Reserved
0	IDLE_EN	SLEEP mode enable. 1: the device enters SLEEP mode. 0: the device exits SLEEP mode.

4.6.8 Clock control register (SYS_CLK_CFG)

XRAM_SFR Address: 0x2028

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	WAIT_MODE	PLL_CLK_SEL	
Read/Write	-	-	-	-	-	Read/Write	Read/Write	
Po initial value	-	-	-	-	-	0	1	0

Bit No.	Symbol	Description
2	WAIT_MODE	WAIT mode enabled 1: the device enters WAIT mode 0: the device exits WAIT mode
1 to 0	PLL_CLK_SEL	PLL Clock frequency divider selects register 00: 24MHz 01: 16MHz 10: 12MHz 11: 8MHz

4.6.9 LVDT Rising/Falling Voltage Interrupt Status Register (INT_POBO_STAT)

XRAM_SFR Address: 0x2029

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	INT_PO_STAT	INT_BO_STAT
Read/Write	-	-	-	-	-	-	Read/Write	Read/Write
Po initial value	-	-	-	-	-	-	0	0

Bit No.	Symbol	Description
1	INT_PO_STAT	Lvdt rising voltage interrupt status 1: rising voltage interrupt is valid 0: rising voltage interrupt is invalid
0	INT_BO_STAT	Lvdt falling voltage interrupt status 1: falling voltage interrupt is valid 0: falling voltage interrupt is invalid

4.6.10 WDT Test Mode Register (WDT_TEST_EN)

XRAM_SFR Address: 0x2060

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	--	--
Read/Write	-	-	-	-	-	-	Read/Write	Read/Write
Po initial value	-	-	-	-	-	-	0	0

Bit No.	Symbol	Description
1~0	--	WDT test mode register Set value 0x02 to configure WDT_CLK_SEL, other values cannot configure the register.

4.6.11 WDT Clock Selection Register (WDT_CLK_SEL)

XRAM_SFR Address: 0x2061

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	--
Read/Write	-	-	-	-	-	-	-	Read/Write
Po initial value	-	-	-	-	-	-	-	0

Bit No.	Symbol	Description
0	--	WDT clock selection register 0: selects the RC32KHz 1: selects the 24MHz clock (only used for testing WDT)

4.6.12 WDT Overflow Configuration Register (WDT_CTRL)

XRAM_SFR Address: 0x2068

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	WDT_TIME_SEL		
Read/Write	-	-	-	-	-	Read/Write		
Po initial value	-	-	-	-	-	0		

Bit No.	Symbol	Description
2~0	WDT_TIME_SEL	WDT overflow configuration register settings: 0x00：18ms； 0x01：36ms； 0x02：72ms； 0x03：144ms； 0x04：288ms； 0x05：576ms； 0x06：1152ms；0x07：2304ms；

WDT uses RC32768Hz to implement the timer function ranging from 18ms to 2.3s, of which time is controlled by WDT_CTRL.

4.6.13 WDT Enable Configuration Register (WDT_EN)

XRAM_SFR address: 0x2069

Bit No.	7	6	5	4	3	2	1	0
Symbol	WDT_EN							
Read/Write	Read/Write							
Po initial value	0							

Bit No.	Symbol	Description
7~0	WDT_EN	WDT Enable configuration register: when the value is set as 0x55, WDT is disabled.

Write 0x55 to disable the WDT, and write other values to enable it. WDT keeps working once reset. WDT clear is completed through writing WDT_CTRL. No matter what value is written to this register, it will clear WDT.

4.6.14 WDT/Timer2 Interrupt Status Register (INT_PE_STAT)

Address: 0x91

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	INT_WDT_STAT	INT_TIMER2_STAT
Read/Write	-	-	-	-	-	-	Read/Write	Read/Write
Po initial value	-	-	-	-	-	-	0	0

Bit No.	Symbol	Description
1	INT_WDT_STAT	WDT interrupt status flag bit: write 0 to clear; write WDT_CTRL can also clear. 1: Valid Interrupt

		0: Invalid Interrupt
0	INT_TIMER2_STAT	TIMER2 interrupt status flag bit: write 0 to clear; Write TIMER2_CFG can also to clear. 1: valid interrupt 0: Invalid Interrupt

5 GPIO

5.1 Description

Some pins of GPIO lines multiplex the peripherals functions of the device. It can not be configured with many functions at the same time, otherwise it causes functional chaos. IIC with open drain output, requires an external pull-up resistor.

The pull-up and pull-down resistance value are 10K for all I/Os.

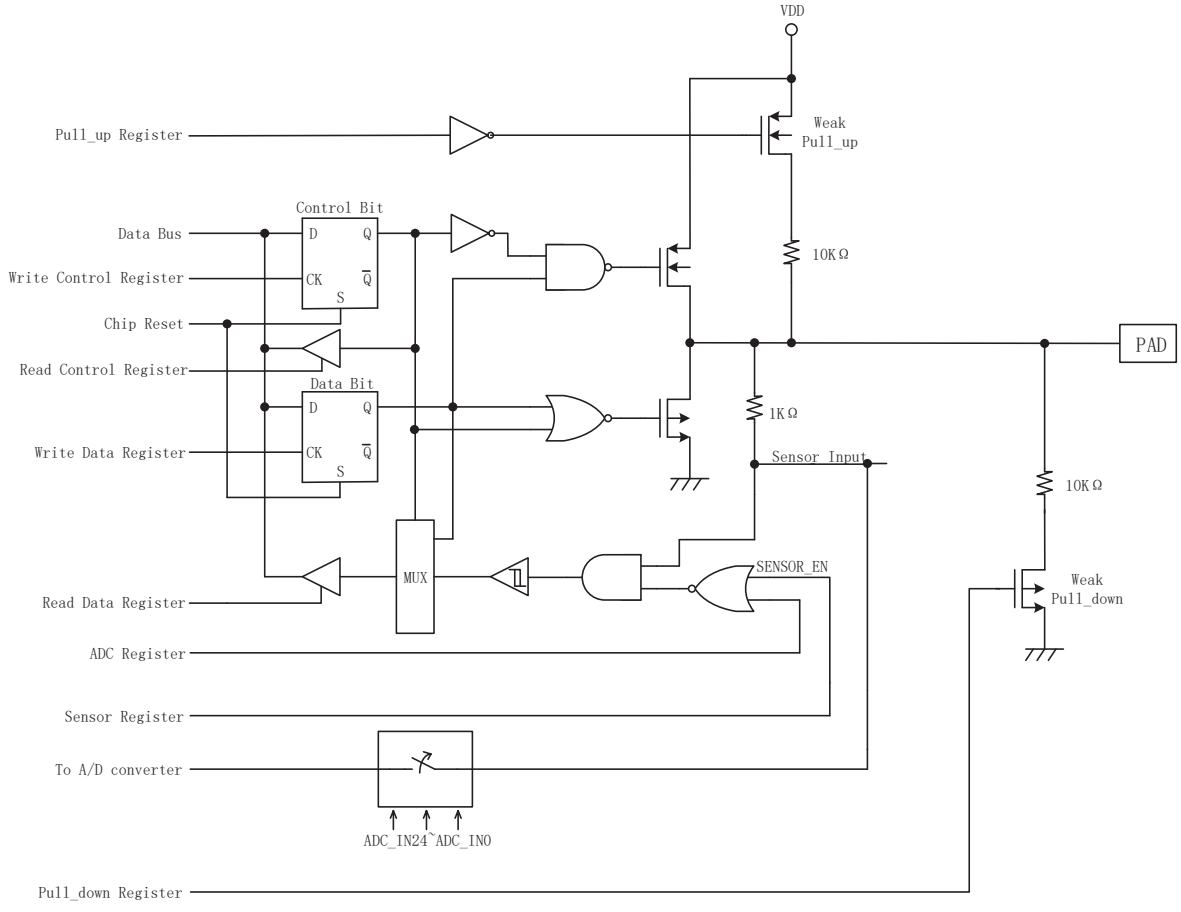


Figure 0.1 GPIO Structure Diagram

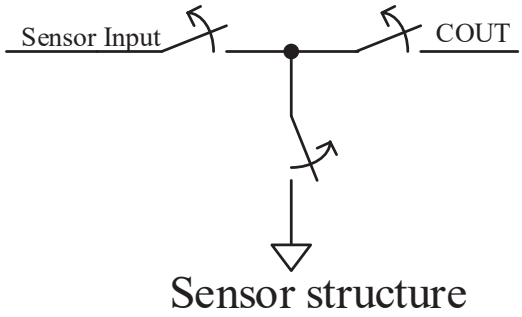


Figure 0.1 SNS IO Structure

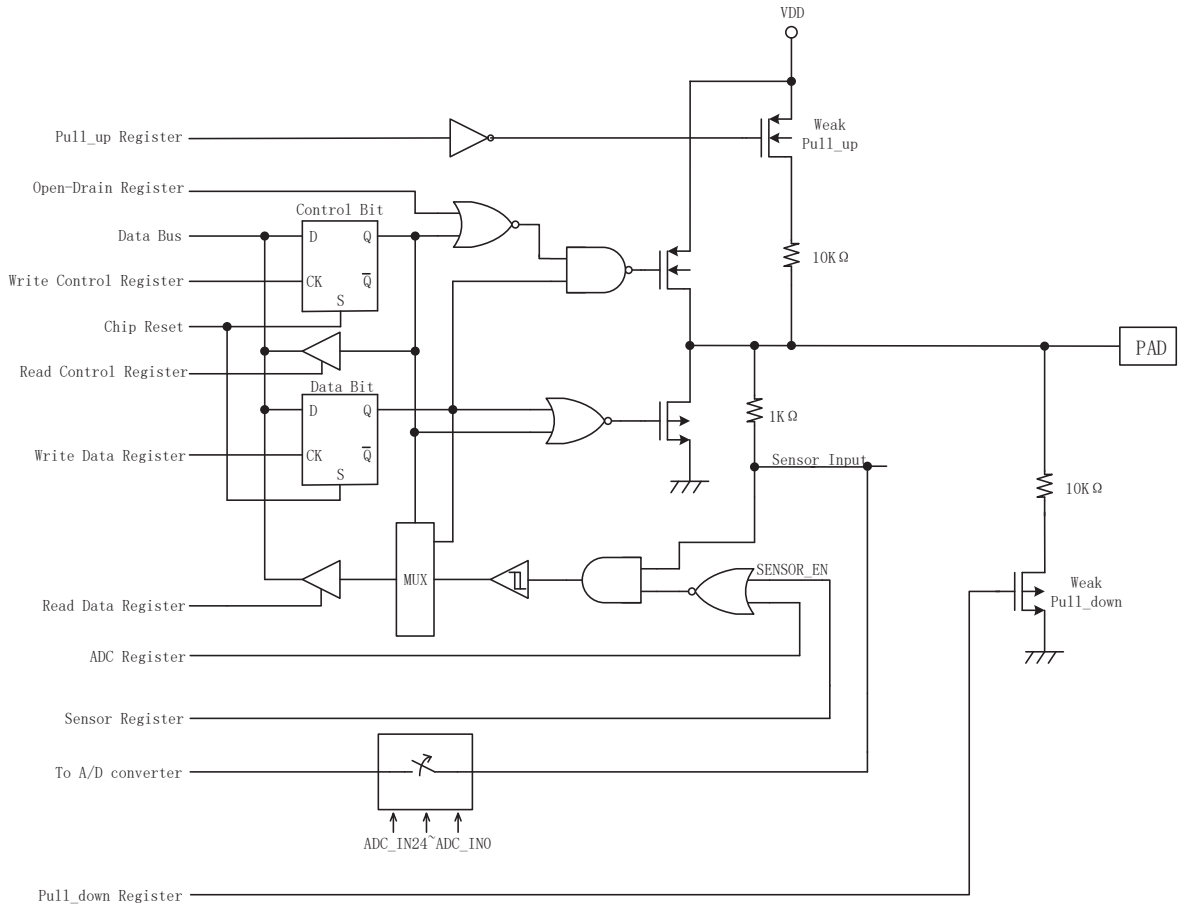


Figure 0.2 IO with Open Drain Output Structure

TRISX (direction register): TRISX set 1 to configure the corresponding pin as input, while set 0 to configure that as output.

DATAX_IN (data input register): Read-only. The read value represents the current level of the present I/O (input).

DATAX_OUT (data output register): configure the output current level when I/O of PX group as GPIO, which reads out the configuration output value.

PULLUP_PX (pull-up resistor enable register): PULLUP_PX set 1 to enable the pull-up resistor on the corresponding pin, set 0 to disable that.

PULLDOWN_PX (pull-down resistor enable register): PULLDOWN_PX set 1 to enable the pull-down resistor on the corresponding pin, set 0 to disable that.

ODRAIN_EN: ODRAIN_EN set 1 to enable the open drain output on the corresponding pin, set 0 to disable that function. After enabling IIC, it automatically turn on the open drain output. It is recommended to use external pull-up resistor for IIC/UART.

5.2 GPIO Registers

5.2.1 Data Registers

5.2.1.1 PA Data Input Register (DATAA_IN)

Address: 0x84

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	-	-	Read	Read	Read	Read	Read	Read
Po Initial Value	-	-	state	state	state	state	state	state

Bit No.	Symbol	Description
5~0	--	PA data input register. Read-out value is the current level of the present IO (input).

PA data register always reads out 1 for sensor/adc functions; for other functions, it reads out the current level of PAD.

5.2.1.2 PB Data Input Register(DATAB_IN）

Address: 0x85

Bit No.	7	6	5	4	3	2	1	0
Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Read/Write	Read	Read	Read	Read	Read	Read	Read	Read
Po Initial Value	state	state	state	state	state	state	state	state

Bit No.	Symbol	Description
7~0	--	PB data input register. Read-out value is the current level of the present IO (input).

PB data input register always reads out 1 for sensor/adc functions; for other functions, it reads out the current level of PBD.

5.2.1.3 PC Data Input Register (DATAC_IN)

Address: 0x86

Bit No.	7	6	5	4	3	2	1	0
Symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Read/Write	Read	Read	Read	Read	Read	Read	Read	Read
Po Initial Value	state	state	state	state	state	state	state	state

Bit No.	Symbol	Description
7~0	--	PC data input register. Read-out value is the current level of the present IO (input).

PC data input register always reads out 1 for sensor/adc functions; for other functions, it reads out the current level of PCD.

5.2.1.4 PD Data Input Register (DATAD_IN)

Address: 0x8E

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	PD4	PD3	PD2	PD1	PD0
Read/Write	-	-	-	Read	Read	Read	Read	Read
Po Initial Value	-	-	-	state	state	state	state	state

Bit No.	Symbol	Description
4~0	--	PD data input register. Read-out value is the current level of the present IO (input).

PD data input register always reads out 1 for sensor/adc functions; for other functions, it reads out the current level of PDD.

5.2.1.5 PA Data Output Register (DATAA_OUT)

Address: 0xB0

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	-	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write

Po Initial Value	-	-	0	0	0	0	0	0

Bit No.	Symbol	Description
5~0	--	PA data output register. Configure the output current level of PA group IOs as GPIOs, which reads out the configuration output value.

5.2.1.6 PB Data Output Register (DATAB_OUT)

Address: 0x80

Bit No.	7	6	5	4	3	2	1	0
Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7~0	--	PB data output register. Configure the output current level of PB group IOs as GPIOs, which reads out the configuration output value.

5.2.1.7 PC Data Output Register (DATAC_OUT)

Address: 0x90

Bit No.	7	6	5	4	3	2	1	0
Symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7~0	--	PC data output register. Configure the output current level of PC group IOs as GPIOs, which reads out the configuration output value.

5.2.1.8 PD Data Output Register (DATAD_OUT)

Address: 0x98

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	PD4	PD3	PD2	PD1	PD0
Read/Write	-	-	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	-	-	0	0	0	0	0

Bit No.	Symbol	Description
4~0	--	PD data output register. Configure the output current level of PD group IOs as GPIOs, which reads out the configuration output value.

5.2.2 Output Control Hold Register When Reset (PD_HOLD_EN)

XRAM_SFR Address: 0x202F

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	-
Read/Write	-	-	-	-	-	-	-	Read/Write
Po Initial Value	-	-	-	-	-	-	-	0

Bit No.	Symbol	Description
0	--	Whether DATAD_OUT/TRISD holds output when reset. 1: reset configuration holding 0: reset configuration failure

5.2.3 Direction Registers

5.2.3.1 PA Direction Register(TRISA)

XRAM_SFR Address: 0x2030

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	-	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial	-	-	1	1	1	1	1	1

Value								

Bit No.	Symbol	Description
5~0	--	PA direction register. 0: output 1: input

5.2.3.2 PB Direction Register (TRISB)

XRAM_SFR Address: 0x2031

Bit No.	7	6	5	4	3	2	1	0
Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	1	1	1	1	1	1	1	1

Bit No.	Symbol	Description
7~0	--	PB direction register. 0: output 1: input

5.2.3.3 PC Direction Register (TRISC)

XRAM_SFR Address: 0x2032

Bit No.	7	6	5	4	3	2	1	0
Symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	1	1	1	1	1	1	1	1

Bit No.	Symbol	Description
7~0	--	PC direction register. 0: output 1: input

5.2.3.4 PD Direction Register (TRISD)

XRAM_SFR Address: 0x2033

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	PD4	PD3	PD2	PD1	PD0
Read/Write	-	-	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	-	-	1	1	1	1	1

Bit No.	Symbol	Description
4~0	--	PD direction register. 0: output 1: input

5.2.4 Pull-up Resistor Enable Registers

5.2.4.1 PA Pull-up Resistor Control Register (PULLUP_PA)

XRAM_SFR Address: 0x2034

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	-	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	-	0	0	0	0	0	0

Bit No.	Symbol	Description
5~0	--	PA port pull-up resistor control register. 1: Enabled 0: Disabled

5.2.4.2 PB Pull-up Resistor Control Register (PULLUP_PB)

XRAM_SFR Address: 0x2035

Bit No.	7	6	5	4	3	2	1	0
Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7~0	--	PB port pull-up resistor control register. 1: Enabled 0: Disabled

5.2.4.3 PC Pull-up Resistor Control Register (PULLUP_PC)

XRAM_SFR Address: 0x2036

Bit No.	7	6	5	4	3	2	1	0
Symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7~0	--	PC port pull-up resistor control register. 1: Enabled 0: Disabled

5.2.4.4 PD Pull-up Resistor Control Register (PULLUP_PD)

XRAM_SFR Address: 0x2037

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	PD4	PD3	PD2	PD1	PD0
Read/Write	-	-	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	-	-	0	0	0	0	0

Bit No.	Symbol	Description
4~0	--	PD port pull-up resistor control register. 1: Enabled 0: Disabled

5.2.5 Pull-down Resistor Enable Registers

5.2.5.1 PA Pull-down Resistor Control Register (PULLDOWN_PA)

XRAM_SFR Address: 0x2038

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	PA5	PA4	PA3	PA2	PA1	PA0
Read/Write	-	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	-	0	0	0	0	0	0

Bit No.	Symbol	Description
5~0	--	PA pull-down resistor control register. 1: Enabled 0: Disabled

5.2.5.2 PB Pull-down Resistor Control Register (PULLDOWN_PB)

XRAM_SFR Address: 0x2039

Bit No.	7	6	5	4	3	2	1	0
Symbol	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7~0	--	PB pull-down resistor control register. 1: Enabled 0: Disabled

5.2.5.3 PC Pull-down Resistor Control Register (PULLDOWN_PC)

XRAM_SFR Address: 0x203A

Bit No.	7	6	5	4	3	2	1	0
Symbol	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Read/Writ	Read/Writ	Read/Writ	Read/Writ	Read/Writ	Read/Writ	Read/Writ	Read/Writ	Read/Writ

e	e	e	e	e	e	e	e	e
Po Initial Value	0	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7~0	--	PC pull-down resistor control register. 1: Enabled 0: Disabled

5.2.5.4 PD Pull-down Resistor Control Register (PULLDOWN_PD)

XRAM_SFR Address: 0x203B

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	PD4	PD3	PD2	PD1	PD0
Read/Write	-	-	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	-	-	0	0	0	0	0

Bit No.	Symbol	Description
4~0	--	PD pull-down resistor control register. 1: Enabled 0: Disabled

5.2.6 Open Drain Output Enable Register(ODRAIN_EN)

XRAM_SFR Address: 0x204A

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	-
Read/Write	-	-	-	-	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	-	-	-	0	0	0	0

Bit No.	Symbol	Description
3	--	PA1 Open Drain Output Enable Register. 1: Open drain output 0: CMOS output
2	--	PA2 Open Drain Output Enable Register. 1: Open drain output 0: CMOS output

1	--	PB6 Open Drain Output Enable Register. 1: Open drain output 0: CMOS output
0		PB7 Open Drain Output Enable Register. 1: Open drain output 0: CMOS output

5.3 I/O Multiplexing Tables

5.3.1 SPI I/O Multiplexing (SPI_IO_SEL)

XRAM_SFR address: 0x2044

Bit No.	Symbol	I/O Description	I/O Multiplexing
7~4	--	--	--
3	MISO_IO_SEL	SPI master input	MISO selection 0: PA2; 1: PC3
2	MOSI_IO_SEL	SPI master output	MOSI selection 0: PA1; 1: PC4
1	SCK_IO_SEL	SPI clock line	SCK selection 0: PA3; 1: PC2
0	CS_IO_SEL	SPI chip select signal	CS selection 0: PD0; 1:PC5

5.3.2 UART I/O Multiplexing (UART_IO_SEL)

XRAM_SFR address: 0x2045

Bit No.	Symbol	I/O Description	I/O Multiplexing
7~6	--	--	--
5~3	UART_RXD_SEL	Transmit Serial Data	UART_RXD Selection 0: PA2; 1: PB1; 2: PB5; 3: PB7; 4: PC1
2~0	UART_TXD_SEL	Transmit Serial Data	UART_TXD selection: 0: PA1; 1: PB2;

			2: PB4; 3: PB6; 4: PC0

5.3.3 SCI I/O Multiplexing (SCI_IO_SEL)

XRAM_SFR address: 0x2046

Bit No.	Symbol	I/O Description	I/O Multiplexing
7~6	--	--	--
5~3	SCI_RXD_SEL	SCI receive	SCI_RXD selection 0：PA2；1：PB1；2：PB5；3：PB7； 4：PC1
2~0	SCI_TXD_SEL	SCI transmit	SCI_TXD selection 0：PA1；1：PB2；2：PB4；3：PB6； 4：PC0

5.3.4 PWM I/O Multiplexing

- PWM_IO_SEL0

XRAM_SFR address: 0x2047

Bit No.	Symbol	I/O Description	I/O Multiplexing
7	PWM_ETR_EN	PWM External Trigger Enable	PWM_ETR I/O Enable 0：Disable；1：Enable
6	PWM_ETR	PWM External Trigger	PWM_ETR Selection 0：PA3；1：PD4
5~4	PWM0_CH3_SEL	PWM0 Channel 2	PWM0_CH3 selection 0：PD4；1：PB2；2：PC4
3~2	PWM0_CH2_SEL	PWM0 Channel 1	PWM0_CH2 selection 0：PB0；1：PB6；2：PC6
1~0	PWM0_CH1_SEL	PWM0 Channel 0	PWM0_CH1 selection 0：PB1；1：PB7；2：PC7

- PWM_IO_SEL1

XRAM_SFR address: 0x2048

Bit No.	Symbol	I/O Description	I/O Multiplexing
7	PWM_BRK_EN	PWM Break I/O Enable	PWM_BRK Enable

			0: Disable；1: Enable
6~5	PWM_BRK	PWM Break I/O	PWM_BRK selection 0: PA1；1: PB0；2: PC6
4	PWM0_CH3N_SEL	PWM0 Complementary Channel 2	PWM0_CH3N selection 0: PA5；1: PD3
3	PWM0_CH2N_SEL	PWM0 Complementary Channel 2	PWM0_CH2N selection 0：PA4；1：PD2
2	PWM0_CH1N_SEL	PWM0 Complementary Channel 2	PWM0_CH1N selection 0：PD1；1：PB5
1	PWM1_CH2_SEL	PWM1Channel 1	PWM1_CH2 selection 0：PD1；1：PC2
0	PWM1_CH1_SEL	PWM1Channel 0	PWM1_CH1 selection 0：PA3；1：PD0

5.3.5 IIC I/O Multiplexing (IIC_IO_SEL)

XRAM_SFR address: 0x2049

Bit No.	Symbol	I/O Description	I/O Multiplexing
7~4	--		--
3	IIC_AFIL_SEL	IIC Analog Filter	IIC I/O analog filter selection enable 1: Select Analog Filter; 0: Do not select Analog filter
2	IIC_DFIL_SEL	IIC Digital Filter	IIC I/O Digital filter selection enable 1: Select digital filter; 0: Do not select digital filter.
1	'IICSDA_IO_SEL	IIC serial data line	IIC_SDA selection 0: PA1; 1: PB6
0	'IICSCL_IO_SEL	IIC serial data line	IIC_SCL Selection 0: PA2; 1: PB7

5.4 GPIO Configuration Flow Diagram

To configure IOs as GPIO requires corresponding settings for the 3 registers below.

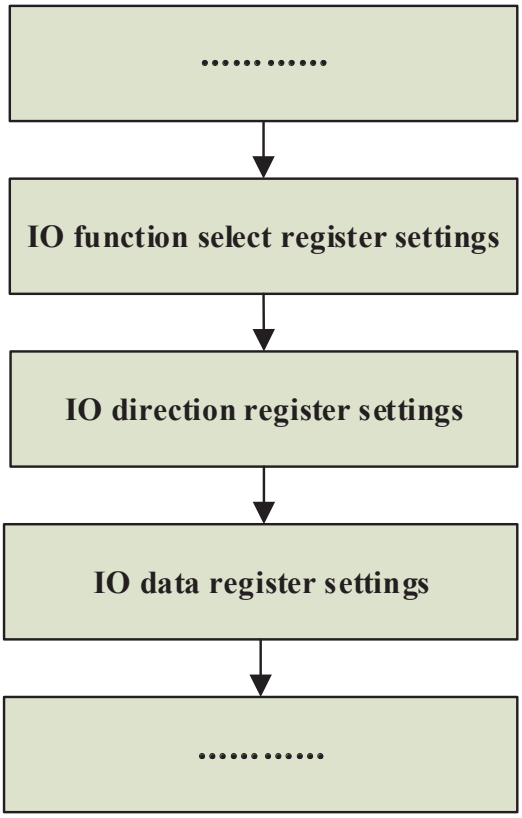


Figure 0.4 IO Configuration Flow Diagram

Note:

By default, the typical I/O current driver is 15mA and the sink current driver is 50mA @5V VCC.

6 Interrupt

The system has 18 interrupts, such as IIC(S), External Interrupt0, External Interrupt1, External Interrupt2, External Interrupt3, Timer0, Timer1, Timer2, RTC, UART, SCI, CSD, ADC, LVDT, INT_WDT, SPI, PWM and PWM1. INT_WDT and Timer2 are multiplexed with one interrupt request flag, which can be set from software or hardware.

6.1 Interrupt Sources and Entries

When the MCU generates a reset signal, the program begins to execute from 0x0000. When an interrupt signal is generated, the program jumps to the address of interrupt vector routine to execute the interrupt service routine.

Interrupt Sources	Conditions	Interrupt Flag		Enable Control	Priority control	Interrupt vector address	Trigger Type	Interrupt No.	Flag Clear Method	Wake up low consumption
INT0	External interrupt 0 current level means the configuration of Trigger Polarity.	IE0	TCON[1]	IEN0[0]	IPL0[0]	0x0003	Falling edge	0	Must be cleared by user	YES
Timer0	Timer0 Overflow	TF0	TCON[5]	IEN0[1]	IPL0[1]	0x000B	High level	1	Must be cleared by user	NO
INT1	External interrupt 1 current level means the configuration of Trigger Polarity.	IE1	TCON[3]	IEN0[2]	IPL0[2]	0x0013	Falling edge	2	Must be cleared by user	YES
Timer1	Timer1 overflow	TF1	TCON[7]	IEN0[3]	IPL0[3]	0x001B	High level	3	Must be cleared by user	NO
INT2	External interrupt 2 current level means the configuration of Trigger Polarity.	IE2	IRCON1[2]	IEN1[2]	IPL1[2]	0x004B	Falling edge	9	Must be cleared by user	YES
INT3	External interrupt 3 current level means the	IE3	IRCON1[3]	IEN1[3]	IPL1[3]	0x0053	Falling edge	10	Must be cleared by user	YES

	configuration of Trigger Polarity.									
ADC	ADC conversion completed	IE4	IRCON1[4]	IEN1[4]	IPL1[4]	0x005B	Falling edge	11	Must be cleared by user	YES
CSD	Conversion completed	IE5	IRCON1[5]	IEN1[5]	IPL1[5]	0x0063	Falling edge	12	Must be cleared by user	YES
SPI	Received or transmitted	IE6	IRCON1[6]	IEN1[6]	IPL1[6]	0x006B	Falling edge	13	Must be cleared by user	NO
WDT/ Timer2	WDT/Timer2 overflow	IE7	IRCON1[7]	IEN1[7]	IPL1[7]	0x0073	Falling edge	14	Must be cleared by user	YES
LVDT	VCC lower than LVDT	IE8	IRCON2[0]	IEN2[0]	IPL2[0]	0x007B	Falling edge	15	Must be cleared by user	YES
UART	Received or transmitted and error	IE9	IRCON2[1]	IEN2[1]	IPL2[1]	0x0083	Falling edge	16	Must be cleared by user	YES
SCI	Received or transmitted and error	IE10	IRCON2[2]	IEN2[2]	IPL2[2]	0x008B	Falling edge	17	Must be cleared by user	YES
RTC	RTC overflow	IE11	IRCON2[3]	IEN2[3]	IPL2[3]	0x0093	Falling edge	18	Must be cleared by user	YES
PWM0	PWM0 Overflow, output compare, input capture	IE12	IRCON2[4]	IEN2[4]	IPL2[4]	0x009B	Falling edge	19	Must be cleared by user	YES
PWM1	PWM1 Overflow, output compare, input capture	IE13	IRCON2[5]	IEN2[5]	IPL2[5]	0x00A3	Falling edge	20	Must be cleared by user	YES

IIC	Slave address match, Received or transmitted	IE14	IRCON2[6]	IEN2[6]	IPL2[6]	0x00AB	Falling edge	21	Must be cleared by user	YES

Table 0.1 Interrupt Information

WDT and Timer2 both use this interrupt number, which can be distinguished by interrupt flag.

6.2 Interrupt Functions

6.2.1 Interrupt Response

When an interrupt request is generated, CPU determines the kind of interrupt according to the interrupt service routine (ISR) and executes ISR completely, unless an interrupt source with higher priority level requests the interrupt. After each ISR, there is a return from interrupt command (RETI). CPU executes RETI command, then continues to execute the program before the interrupt.
ISR can only be requested by an interrupt with higher priority level, which means low priority ISR will be requested to interrupt by high priority interrupt.
BS9000AMxx responses to interrupt request only after executing the present command. If the present command is RETI or accessing IPL and IEN registers, it requires to execute another more instruction before responding to the interrupt request.

6.2.2 Interrupt Handling

Each interrupt source can be individually enabled/disabled through the interrupt enable register. Besides, interrupt request has its own flag bit register, which can trigger interrupt not only by hardware but also via software writing.
BS9000AMxx has two interrupt priority levels, including interrupt level and default priority level. Interrupt level (including highest level, high level and low level) is prior to the default priority level.
Before CPU enters ISR, the default selection of many interrupt sources is decided by natural priority level. Generally, lower bits of interrupt enable registers have a higher priority level. When programmed as higher priority level, these bits will be handled first by the interrupt source, even though others may have a higher natural priority. This means programmed priority level is higher than natural priority level. When the programmed priority levels are the same, it is up to natural priority level to decide the interrupt execution order.
Some interrupt flag bits will be cleared automatically by software after the interrupt response, including Timer0/1 and all external interrupt sources. Other interrupt request flag bits must be cleared by software in the ISR.
CPU has a very short interrupt response time. It takes CPU 2 clock cycles from interrupt request flagged to the start of the execution of ISR.

6.2.3 Interrupt Sampling

Internal modules such as internal timer and serial data generate the interrupt request through their own interrupt flag bit in SFR. When the first clock period (C1) of each instruction cycle completes, it begins to sample for external interrupt on the rising edge of the clock.

To ensure that edge-triggered interrupts are detected, the corresponding port is first kept high for two clocks

To ensure that edge-triggered interrupts are detected, the corresponding port is first kept high for two clocks, then another 2 cycles' low level. The diagram below is the time sequence for interrupt sampling.

The interrupt response time is decided by the present status of system. The fastest response time is 5 instruction cycles: 1 cycle for detecting interrupt request and the other 4 cycles for executing LCALL into ISR.

When system executes RETI command and the MUL or DIV command is waiting in line, it has the longest waiting time, which is 13 instruction cycles, including 1 cycle for detecting interrupt request, 3 cycles for completing RETI command, 5 for DIV or MUL command and 4 for LCALL into ISR. In this case, the response time is 13 clock cycles.

6.3 Interrupt Wakeup

In wait mode, analog clock is not off, only disable F_sys_clk to achieve low consumption mode. In this mode, modules Timer0/1, SPI and CPU do not work, but other modules can be configured to work, which can be waken up by any kinds of interrupt (including int_wdt) except Timer0/1 and SPI. After the wake-up, CPU can work directly.

In sleep mode, disable OSC1M, PLL and other analog modules to save the power, which can be waken up by interrupts such as int_wdt, int_Timer2, int_rtc, int_ext0, int_ext1, int_ext2, int_ext3, int_iic, int_sci or reset.

If many interrupts occurs in the sleep mode, enable the clock at the first interrupt, wait for the clock to be stable and then transmit the interrupt signals to MCU at the same time. After the MCU exits sleep mode, it responds to interrupts in the order of interrupt priority.

Notes:

1. The command that configures CPU to enter sleep mode needs not to be conducted with interrupt trigger at the same time, otherwise the command to enter sleep mode will be neglected. It is recommended to use if statement to decide whether to enter sleep mode. Concurrently, after exiting sleep mode, wait at least for 50us, then configure to enter sleep mode again.
2. During the waiting for wake-up, it requires a interrupt signal length longer than 6 cycles of PLL_24Mhz plus 1 cycle of F_sys_clk, otherwise the interrupt signal may be neglected.

6.4 Interrupt Registers

6.4.1 Interrupt Enable Register 0(IEN0)

Address: 0xA8

Bit No.	7	6	5	4	3	2	1	0
Symbol	EA	-	ET2	-	ET1	EX1	ET0	EX0
Read/ Write	Read/ Write	-	Read/ Write	-	Read/ Write	Read/ Write	Read/ Write	Read/ Write
Po Initial Value	0	-	0	-	0	0	0	0

Bit No.	Symbol	Description
7	EA	Interrupt Permit bit. 0: Mask all interrupts (EA is prior to the interrupt enable bit of interrupt sources). 1: enable interrupt. Whether permitting the interrupt request of every interrupt source or not depends on its own Permit bit.
6	--	Reserved
5	ET2	Timer 2 Interrupt enable bit. 0: Disabled 1: Enabled
4	--	Reserved
3	ET1	Timer 1 Interrupt enable bit. 0: Disabled 1: Enabled
2	EX1	INT_EXT1 Permit bit 0: Disabled 1: Enabled
1	ET0	Timer 0 Interrupt enable bit. 0: Disabled 1: Enabled
0	EX0	INT_EXT0 Permit bit 0: Disabled 1: Enabled

6.4.2 Interrupt Priority Register 0 (IPL0)

Address: 0xB8

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	PT1	PX2	PT0	PX0
Read/Write	-	-	-	-	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	-	-	-	0	0	0	0

Bit No.	Symbol	Description
7~6	-	Reserved bit.
5	PT2	Timer2 Interrupt Priority Register. 0: selects low priority level. 1: selects high priority level.
4	–	Reserved bit.
3	PT1	Timer1 Interrupt Priority Register. 0: selects low priority level. 1: selects high priority level.
2	PX2	INT_EXT1 Interrupt Priority Register. 0: selects low priority level. 1: selects high priority level.
1	PT0	Timer0 Interrupt Priority Register. 0: selects low priority level. 1: selects high priority level.
0	PX0	INT_EXT0 Interrupt Priority Register. 0: selects low priority level. 1: selects high priority level.

6.4.3 Interrupt Flag Register (TCON)

Address: 0x88

Bit No.	7	6	5	4	3	2	1	0
Symbol	TF1	TR1	TF0	TR0	IE1	-	IE0	-
Read/Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	Read/ Write	-	Read/ Write	-
Po Initial Value	0	0	0	0	0	-	0	-

Bit No.	Symbol	Description

7	TF1	Timer1 overflow flag. When Timer1 occurs overflow, hardware is set 1, or the TH0 of Timer0 overflows in mode 3.
6	TR1	Timer1 enable, set to 1 to enable Timer1 or enable TH0 to count in Time0 mode 3.
5	TF0	TImer0 overflow flag bit. When Timer0 overflows, hardware is set to 1.
4	TR0	Timer0 enable. 1: enable Timer0 0: disable Timer0
3	IE1	External interrupt 1 flag bit. Set hardware to 1 to clear software.
2	--	Reserved.
1	IE0	External interrupt 0 flag bit. Set hardware to 1 to clear software.

6.4.4 Interrupt Enable Register 1 (IEN1)

Address: 0xE6

Bit No.	7	6	5	4	3	2	1	0
Symbol	WDT_T2_IE	SPI_IE	CSD_IE	ADC_IE	EX3_IE	EX2_IE	-	-
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	-	-
Po Initial Value	0	0	0	0	0	0	-	-

Bit No.	Symbol	Description
7	WDT_T2_IE	WDT/Timer2 interrupt enable. 1: enabled. 0: disabled.
6	SPI_IE	SPI interrupt enable. 1: enabled. 0: disabled.
5	CSD_IE	CSD interrupt enable. 1: enabled. 0: disabled.
4	ADC_IE	ADC interrupt enable. 1: enabled. 0: disabled.
3	EX3_IE	EX3_IE interrupt enable. 1: enabled. 0: disabled.

2	EX2_IE	EX2_IE interrupt enable. 1: enabled. 0: disabled.
1~0	--	Reserved.

6.4.5 Interrupt Priority Register 1 (IPL1)

Address: 0xF6

Bit No.	7	6	5	4	3	2	1	0
Symbol	WDT_T2_IP	SPI_IP	CSD_IP	ADC_IP	EX3_IP	EX2_IP	-	-
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	-	-
Po Initial Value	0	0	0	0	0	0	-	-

Bit No.	Symbol	Description
7	WDT_T2_IP	WDT/Timer 2 Interrupt Priority level. 0: low priority level. 1: high priority level.
6	SPI_IP	SPI Interrupt Priority level. 0: low priority level. 1: high priority level.
5	CSD_IP	CSD Interrupt Priority level. 0: low priority level. 1: high priority level.
4	ADC_IP	ADC Interrupt Priority level. 0: low priority level. 1: high priority level.
3	EX3_IP	EX3_IP Interrupt Priority level. 0: low priority level. 1: high priority level.
2	EX2_IP	EX2_IP Interrupt Priority level. 0: low priority level. 1: high priority level.
1~0	--	Reserved

6.4.6 Interrupt Flag Register 1(IRCON1)

Address: 0xF1

Bit No.	7	6	5	4	3	2	1	0
Symbol	WDT_T2_IF	SPI_IF	CSD_IF	ADC_IF	EX3_IF	EX2_IF	-	-
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	-	-
Po Initial Value	0	0	0	0	0	0	-	-

Bit No.	Symbol	Description
7	WDT_T2_IF	WDT/Timer2 interrupt flag. 1: flagged. 0: not flagged.
6	SPI_IF	SPI interrupt flag. 1: flagged. 0: not flagged.
5	CSD_IF	CSD interrupt flag. 1: flagged. 0: not flagged.
4	ADC_IF	ADC interrupt flag. 1: flagged. 0: not flagged.
3	EX3_IF	EX3_IF interrupt flag. 1: flagged. 0: not flagged.
2	EX2_IF	EX2_IF interrupt flag. 1: flagged. 0: not flagged.
1~0	--	Reserved

6.4.7 Interrupt Enable Register 2(IEN2)

Address: 0xE7

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	IIC_IE	PWMI_IE	PWM0_IE	RTC_IE	SCI_IE	UART_IE	LVDT_IE
Read/Wri	-	Read/Wri	Read/Wri	Read/Wri	Read/Wri	Read/Wri	Read/Wri	Read/Wri

te		te	te	te	te	te	te	te
Po Initial Value	-	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7~0	--	Reserved.
6	IIC_IE	IIC interrupt enable. 1: enabled. 0: disabled.
5	PWM1_IE	PWM1 interrupt enable. 1: enabled. 0: disabled.
4	PWM0_IE	PWM0 interrupt enable. 1: enabled. 0: disabled.
3	RTC_IE	RTC interrupt enable. 1: enabled. 0: disabled.
2	SCI_IE	SCI interrupt enable. 1: enabled. 0: disabled.
1	UART_IE	UART interrupt enable. 1: enabled. 0: disabled.
0	LVDT_IE	LVDT interrupt enable. 1: enabled. 0: disabled.

6.4.8 Interrupt Priority Register 2 (IPL2)

Address: 0xF4

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	IIC_IP	PWM1_IP	PWM0_IP	RTC_IP	SCI_IP	UART_IP	LVDT_IP
Read/Write	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7	--	Reserved.

6	IIC_IP	IIC Interrupt Priority Register. 0: low priority level. 1: high priority level.
5	PWM1_IP	PWM1Interrupt Priority Register. 0: low priority level. 1: high priority level.
4	PWM0_IP	PWM0 Interrupt Priority Register. 0: low priority level. 1: high priority level.
3	RTC_IP	RTC Interrupt Priority Register. 0: low priority level. 1: high priority level.
2	SCI_IP	SCI Interrupt Priority Register. 0: low priority level. 1: high priority level.
1	UART_IP	UART Interrupt Priority Register. 0: low priority level. 1: high priority level.
0	LVDT_IP	LVDT Interrupt Priority Register. 0: low priority level. 1: high priority level.

6.4.9 Interrupt Flag Register 2 (IRCON2)

Address: 0xE1

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	IIC_IF	PWM1_IF	PWM0_IF	RTC_IF	SCI_IF	UART_IF	LVDT_IF
Read/Write	-	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	-	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7	--	Reserved
6	IIC_IF	IIC interrupt flag. 1: flagged. 0: not flagged.
5	PWM1_IF	PWM1 interrupt flag. 1: flagged.

		0: not flagged.
4	PWM0_IF	PWM0 interrupt flag. 1: flagged. 0: not flagged.
3	RTC_IF	RTC interrupt flag. 1: flagged. 0: not flagged.
2	SCI_IF	SCI interrupt flag. 1: flagged. 0: not flagged.
1	UART_IF	UART interrupt flag. 1: flagged. 0: not flagged.
0	LVDT_IF	LVDT interrupt flag. 1: flagged. 0: not flagged.

6.4.10 INT3 Selection Enable Register 3 (EXT3_INT_SEL3)

XRAM_SFR Address: 0x204B

Bit No.	7	6	5	4
Symbol	INT3_23_IO_SEL	INT3_22_IO_SEL	INT3_21_IO_SEL	INT3_20_IO_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	INT3_19_IO_SEL	INT3_18_IO_SEL	INT3_17_IO_SEL	INT3_16_IO_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7~0	INT3_n_IO_SEL (n=23~16)	INT3_n IO selection enable. 1: enable INT. 0: disable INT.

6.4.11 INT3 Selection Enable Register 2 (EXT3_INT_SEL2)

XRAM_SFR Address: 0x204C

Bit No.	7	6	5	4
Symbol	INT3_15_IO_SEL	INT3_14_IO_SEL	INT3_13_IO_SEL	INT3_12_IO_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	INT3_11_IO_SEL	INT3_10_IO_SEL	INT3_9_IO_SEL	INT3_8_IO_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7~0	INT3_n_IO_SEL (n=15~8)	INT3_n IO selection enable. 1: enable INT. 0: disable INT.

6.4.12 INT3 Selection Enable Register 1 (EXT3_INT_SEL1)

XRAM_SFR Address: 0x204D

Bit No.	7	6	5	4
Symbol	INT3_7_IO_SEL	INT3_6_IO_SEL	INT3_5_IO_SEL	INT3_4_IO_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	INT3_3_IO_SEL	INT3_IO_2_SEL	INT3_1_IO_SEL	INT3_0_IO_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7~0	INT3_n_IO_SEL (n=7~0)	INT3_n IO selection enable. 1: enable INT. 0: disable INT.

6.4.13 INT Selection Enable Register (EXT_INT_SEL)

XRAM_SFR Address: 0x204E

Bit No.	7	6	5	4
Symbol	-	-	-	-
Read/Write	-	-	-	-
Po Initial Value	-	-	-	-
Bit No.	3	2	1	0
Symbol	-	INT2_IO_SEL	INT1_IO_SEL	INT0_IO_SEL
Read/Write	-	Read/Write	Read/Write	Read/Write
Po Initial Value	-	0	0	0

Bit No.	Symbol	Description
2	INT2_IO_SEL	INT2 IO selection enable. 1: enable INT2. 0: disable INT2.
1	INT1_IO_SEL	INT1 IO selection enable. 1: enable INT1. 0: disable INT1.
0	INT0_IO_SEL	INT0 IO selection enable. 1: enable INT0. 0: disable INT0.

6.4.14 External Interrupt Polarity Control Register (EXT_INT_CON)

XRAM_SFR Address: 0x204F

Bit No.	7	6	5	4
Symbol	-	INT3_POLARITY	INT2_POLARITY	
Read/Write	-	Read/Write	Read/Write	Read/Write
Po Initial Value	-	0	0	1
Bit No.	3	2	1	0
Symbol	INT1_POLARITY		INT0_POLARITY	
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	1	0	1

Bit No.	Symbol	Description
6	INT3_POLARITY	External interrupt 3_x trigger polarity selection. 1: rising edge (High level wake-up in low power mode) 0: falling edge (Low level wake-up in low power mode)
5~4	INT2_POLARITY	External interrupt 2 trigger polarity selection: 01: falling edge (Low level wake-up in low power mode) 10: rising edge (High level wake-up in low power mode) 00/11: rising/falling edges (Low level wake-up in low power mode)
3~2	INT1_POLARITY	External interrupt 1 trigger polarity selection: 01: falling edge (Low level wake-up in low power mode) 10: rising edge (High level wake-up in low power mode) 00/11: rising/falling edges (Low level wake-up in low power mode)
1~0	INT0_POLARITY	External interrupt 0 trigger polarity selection: 01: falling edge (Low level wake-up in low power mode) 10: rising edge (High level wake-up in low power mode) 00/11: rising/falling edges (Low level wake-up in low power mode)

Note :

INT3 registers use a common interrupt vector, which can only respond to one external interrupt at the same time.When external interrupts on the rising edge or falling edge of the multiplexing I/Os are enabled, it requires all the enabled external interrupt IOs to be released before responding to the present signal during the detection (when falling edge is triggered, release the high level; when rising edge is triggered, release the low level.)

6.5 External Interrupt Configuration Procedures

External interrupts do not involve the filter handling. They need to configure polarity. For external interrupt INT0/1/2, there are three configurations including rising edge, falling edge and both edges in normal mode. For INT3_0~3_23, it supports rising edge and falling edge in normal mode.
Low-power mode supports high-level and low-level wake-up.
See the configuration for INT0/1/2/3 below.

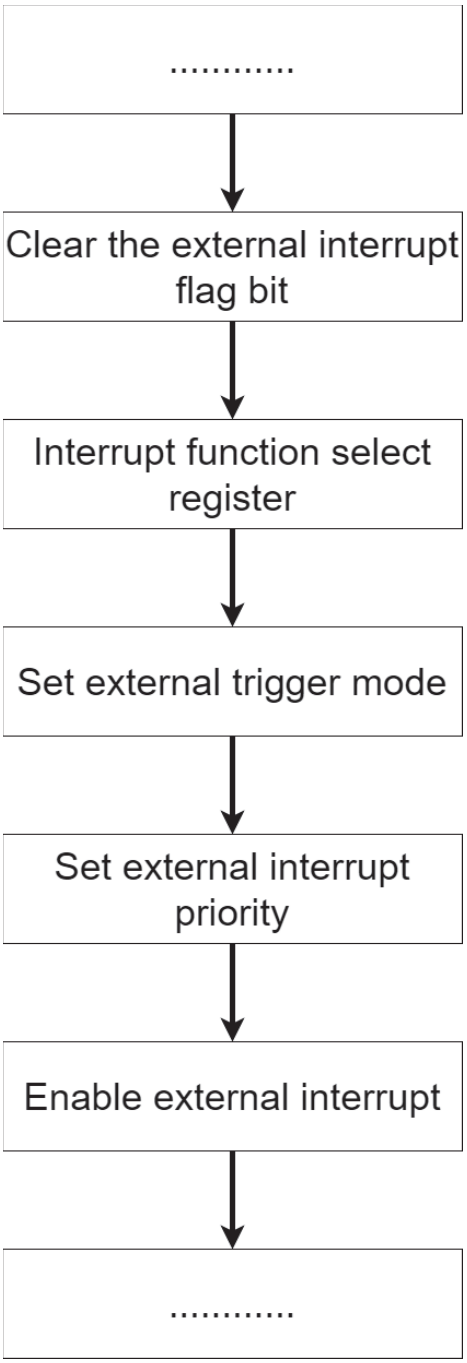


Figure 0.2 INT0/1/2/3 Configuration Flow Chart

First, configure INTx_POLARITY to select polarity. Then, open the interrupt port of the GPIO to enable INTx_IO_SEL.

In both edges mode, if the current level of the present interrupt IO needs to read out, it must to configure TRIS as Input (“1 ”). Determine the current leve by reading out the DATA, which distinguishes between the rising edge and falling edge.

INT3_0~3_23,must configure the corresponding TRIS of the external interrupt as input ('1"), then read out DATA to determine the present level, which distinguishes which IO has been interrupted. (This involves only mechanical keyboard application, usually it rakes a long time for pressing the

buttons.)

Note: INT3_0~3_23 has a common interrupt number for all input signals “ or ” transmitting interrupt signals. Therefore, when many interrupt IOs of INT3_x_IO_SEL are configured validly, make sure only one signal input is valid at the moment, because one of the enabled input signals is within the valid level, changes in others can not trigger interrupt response.

7 Timer

BS9000AMxx has 2 timers (TIMER0, TIMER1) form the core and 2 external timers (TIMER2, RTC). Each Timer contains a 16-bit register that is visible in the form of 2 byte when accessed, including a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The register of Timer2 is a low byte from TIMER2_SET_L and a high byte from TIMER2_SET_H. The register of RTC is a low bye from RTC_SET_L and a high byte from RTC_SET_H.

Timer functions:

- 4 16-bit Timer
- Timer0 connects to the system clock with F_sys_clk /12
- Timer1connects to the system clock with F_sys_clk /12
- Timer2 can select internal RC32K, external oscillator or PLL_24Mhz, and the last two support frequency divider.
- RTC can choose internal RC32K, external oscillator or PLL_24Mhz, and the last two support frequency divider.
- Timer0 supports 8-bit auto-reload timer/counter, 16-bit manual-reload timer/counter.
- Timer1 supports 8-bit auto-reload timer/counter, 16-bit manual-reload timer/counter.
- Timer2 supports 16-bit auto-reload timer and manual-reload timer as well as wake-up from interrupt.
- RTC supports 16-bit auto-reload timer and manual-reload timer as well as wake-up from interrupt.

7.1 Timer0 and Timer1

Set the ET0 bit in the IEN0 register to enable Timer0 interrupt, and set the ET1 bit in the IEN0 to enable the Timer1 interrupt. Set the TR0/1 to enable timer, and decide if the timer overflows through TF0/1. Timer0/1 is controlled by registers TMOD and TCON, they have four operating modes as following:

1. Mode 0: 13-bit timer/counter
2. Mode 1: 16-bit timer/counter
3. Mode 2: auto-reload the 8-bit counter of initial value.
4. Mode 3: two 8-bit timer/counter (only for timer/counter 0)

Mode 0: 13-bit timer logic structure:

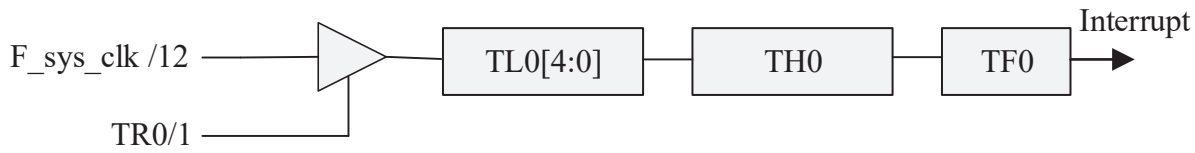


Figure0.1 Mode 0 Logic Structure Diagram

Figure 7.1 shows that Timer0 and Timer1 have the same procedure. In Mode 0, Timer0 is a 13-bit counter that consists of 8 bits of TH0 and low 5 bits of TL0. Timer1 is also a 13-bit counter consisting of 8 bits of TH1 and low 5 bits of TL1. Just ignore the high 3 bits of TL0 and TL1. The enable bit (TR0/TR1) in the TCON register is to control timer ON/OFF.

Timer starts to count for selected system clock source (F_sys_clk/12), when the 13-bit counter accumulates to all 1, the counter will clear to 0 (all 0), and set TF0(or TF1) to 1,. The counter enable is only controlled by TR0/1.

Mode 1: 16-bit timer logic structure:

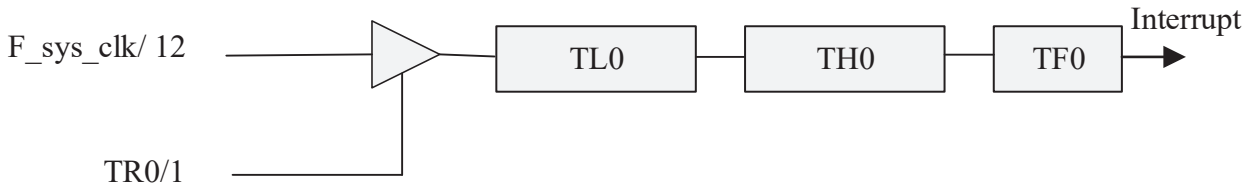


Figure0.2 Mode 1 Logic Structure Diagram

As shown in Figure 7.2, Timer 0 and Timer 1 have the same Mode 1. In mode 1, both Timer0 and

Timer1 are 16-bit counter consisting of 8 bits from TH0 and 8 bits from TL0. When counter accumulates to 0xFFFF, the counter will clear to all 0. Besides, like Mode 0, the counter of Mode 1 is controlled only by TR0/1.

Mode 2: 8-bit timer that auto-reloads the initial value, see its logic structure below:

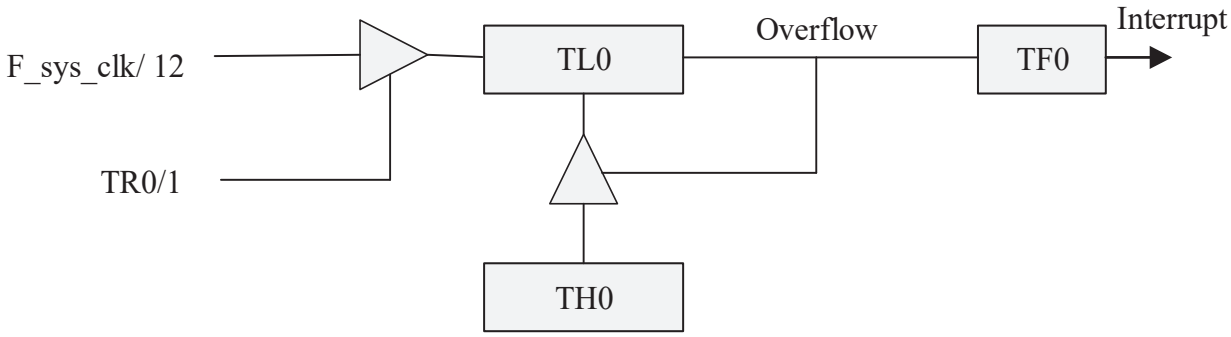


Figure0.3 Mode 2 Logic Structure Diagram

Timer 0 and Timer1 have the same logic in Mode 2. In mode 2, timer is an 8-bit counter with auto-reload initial value function, also know as LSB register (TL0 or TL1), the initial value that needs to auto-reload is stored in the MSB register (TH0 or TH1).

As shown in Figure 7.3, the counter control in mode 2 is the same as in Mode 0 and Mode 1. However, in Mode 2, when TLn accumulates to FFh, the value stored in THn will reload to TLn. The counter is controlled only by TR0/1.

Mode 3: Two 8-bit Timers

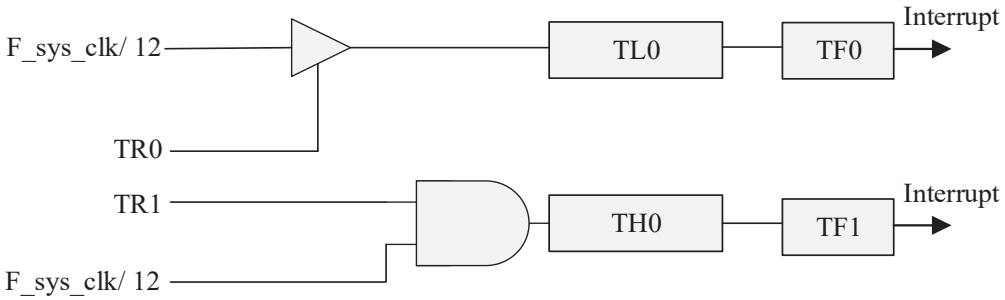


Figure 0.4 Mode 3 Logic Structure Diagram

In Mode 3, Timer0 is two 8-bit timers, then Timer1 stops counting and stores its value. As shown in Figure 7.4, TL0 is an 8-bit register controlled by the control bit of Timer0. The counter uses GATE as enable interface to control the signal transmit and receive of INT_EXT.

TH0 has a individual 8-bit timer. TH0 can only be used to count the clock cycle. The enable bit and

flag bit of Timer1(TR1 and TF1) can serve as the enable bit and flag bit of TH0.
When the Timer0 is in Mode 3, Timer1 will be limited, because Timer0 has used the enable bit (TR1) and interrupt flag bit (TF1) in Timer1. For Timer1. it can still be used to generate the baud rate, and the values in registers TL1 and TH1 are valid.

When Timer0 is in Mode 3, it controls Timer1 through the mode bit from Timer1. To enable Timer1 requires to set the Timer1 as Mode 0 or 1 or 2. To disable Timer1 requires to set its mode to 3. Timer1can serves as a timer (F_sys_clk/12), but it can not generate overflow to interrupt due to the TR1 and TF1 is borrowed by Timer0. When Time0 in Mode 3, the counter enable is controlled by TR0/1.

7.2 Timer2/RTC

RTC module has the same function as Timer2.
The Timer2/RTC modules mainly features a 16-bit counter, which implements its timer function by counting input clock. The counting of Timer2/RTC is based on summation, the counter will trigger interrupt when accumulating to the setting value. Timer2/RTC timer can select internal RC, external oscillator or PLL_24Mhz clock, and the last two clocks support frequency division, which is depended on the clock selection register, computing the time setting range is 0s to 2s.
Enable the Timer2 by configuring the register TIMER2_EN, TIMER2_RLD has auto-reload or manual-reload mode, timer can be set in the registers TIMER2_SET_L and TIMER2_SET_H. RTC has the same configuration as Timer2.
Both Timer2 and RTC support interrupt wake-up from Wait mode and Sleep mode, and require software to clear interrupt flag in the interrupt handler function.
Timer2 and RTC supports two operating modes including single-cycle timing mode and auto-reload mode. No matter what mode it is, it will trigger interrupt after the time is up.

- Single-cycle timing mode: when time is up, the hardware will automatically pull down TIMER2_EN/RTC_EN and stop counting.
- Auto-reload mode: hardware will reload the set value, TIMER2_EN/RTC_EN continues to hold high level, restarting the next counting. Software writes 0 in the register TIMER2_EN/RTC_EN or modify the timer mode in the middle to stop Timer2/RTC.

T_{TIMER2} can be calculated as follows:

$T_{TIMER2}=T_{TIMER2\_CLK}*(\{TIMER2\_SET\_H, TIMER2\_SET\_L\}+1)$

Note:

Any configuration for TIMER2_SET_H, TIMER2_SET_L, TIMER2_CFG can clear the counter.

T_{RTC} can be calculated as follows:

$T_{RTC}=T_{RTC\_CLK}*(\{RTC\_SET\_H, RTC\_SET\_L\}+1)$

Note: any configuration for RTC_SET_H, RTC_SET_L or RTC_CFG can clear the counter.

7.3 Timer Registers

7.3.1 Timer0/1 Registers

7.3.1.1 Timer Control Register (TCON)

Address: 0x88

Bit No.	7	6	5	4	3	2	1	0
Symbol	TF1	TR1	TF0	TR0	IE1	-	IE0	-
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	-	Read/Write	-
Po Initial Value	0	0	0	0	0	-	0	-

Bit No.	Symbol	Description
7	TF1	Timer1 Overflow flag. When Timer1 overflows, hardware sets to 1, or TH0 in the Timer0 overflows in Mode 3.
6	TR1	Timer1 enable. Set 1 to enable Timer1 or enable TH0 in the Time0 to count in Mode 3.
5	TF0	TImer0 overflow flag. When Timer0 overflows, hardware sets to 1.
4	TR0	Timer0 enable. Set to 1 to enable Timer0.
3	IE1	External interrupt 1 flag. Hardware sets to 1, which supports software clearing.
2	--	Reserved
1	IE0	External interrupt 0 flag. Hardware sets to 1, which supports software clearing.
0	--	Reserved

7.3.1.2 Timer Mode Register (TMOD)

Address: 0x89

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	T1	M1		-	T0	M0	
Read/Write	-	Read/Write	Read/Write		-	Read/Write	Read/Write	
Po Initial Value	-	0	0		-	0	0	

Bit No.	Symbol	Description
7	--	Reserved
6	T1	Timer 1 count/timing selection. By default, 0: Timing mode.
5~4	M1	Timer 1 run mode selection. 00: Mode 0, 13-bit timer/counter 01: Mode 1, 16-bit timer.counter 10: Mode 2, 8-bit counter with auto-reload initial value 11: Mode 3 disabled (Timer 1 disabled)
3	--	Reserved
2	T0	Timer 0 count/timing selection. By default, 0: Timing mode.
1~0	M0	Timer 0 run mode selection. 00: Mode 0, 13-bit timer/counter 01: Mode 1, 16-bit timer.counter 10: Mode 2, 8-bit counter with auto-reload initial value 11: Mode 3, two 8-bit timer/counter

7.3.1.3 Timer 0 Counter Low 8 bits (TL0)

Address: 0x8A

Bit No.	7	6	5	4	3	2	1	0
Symbol	TL0[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	TL0[7:0]	Timer 0 counter low 8 bits.

7.3.1.4 Timer 0 Counter High 8 bits (TH0)

Address: 0x8C

Bit No.	7	6	5	4	3	2	1	0
Symbol	TH0[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description

7~0	TH0[7:0]	Timer 0 Counter High 8 bits

7.3.1.5 Timer 1 Counter Low 8 bits (TL1)

Address: 0x8B

Bit No.	7	6	5	4	3	2	1	0
Symbol	TL1[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	TL1[7:0]	Timer 1 Counter Low 8 bits

7.3.1.6 Timer 1 Counter High 8 bits (TH1)

Address: 0x8D

Bit No.	7	6	5	4	3	2	1	0
Symbol	TH1[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	TH1[7:0]	Timer 1 Counter High 8 bits

7.3.2 Timer2/RTC Registers

7.3.2.1 Timer2 Configuration Register (TIMER2_CFG)

XRAM_SFR Address: 0x206A

Bit No.	7~5	6~4	3~2	1	0
Symbol	--	TIMER2_CLK_FRE	TIMER2_CLK_SEL	TIMER2_RLD	TIMER2_EN
Read/Write	--	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	--	0	0	0	0

Bit No.	Symbol	Description
6~4	TIMER2_CLK_FRE	Timer2 XTAL or PLL_24MHz Clock Frequency Division Selection. 000: select the clock frequency/1 001: select the clock frequency/2 010: select the clock frequency/4 011: select the clock frequency/8 100: select the clock frequency/16 101: select the clock frequency/32 110: select the clock frequency/64 111: select the clock frequency/128 No matter in which mode, configure the register will clear the counter during the counting. If the present counting enable is valid, it will refresh the counter.
3~2	TIMER2_CLK_SEL	Timer2 Clock Selection Register 0: select RC32KHz 1: select XTAL 2/3: select PLL_24MHz
1	TIMER2_RLD	Timer2 auto-reload enable register 1: auto-reload mode. 0: manual reload mode.
0	TIMER2_EN	Timer2 counter enable register. 1：start the counting. 0：stop counting. In manual reload mode, hardware will auto-clear the register after the counting and stop the clock. In auto-reload mode, the register holds the enable after the counting, and clock starts to count from 0 again. In either mode, configure the register as 1 will enable the clock to count from 0.

Configure the register TIMER2_CLK_SEL/TIMER2_CLK_FRE to select the operation of Timer2.

Note:

- Check if the clock is enabled.
- PD_XTAL=0x1, disable XTAL clock. PD_XTAL=0x0, enable XTAL. By default, this clock is off.
- In single-cycle timing mode, when the timing is up, it transmits an interrupt signal, then the interrupt falling edge will auto-clear the register.
- In auto-reload mode, when the timing is completed, interrupt signal is sent, enable register holding, the clock goes to the next round of timing.
- Configure the register at any time will reset it, therefore, after the interrupt signal is sent,

the register will auto-clear the counter and the flag. If the configuration is 1 at this time, it will count from 0.

- In auto-reload mode, concerning the accuracy, it may not to configure the timer registers in the interrupt handler functions, because such operations will clear the counter.

7.3.2.2 Timer2 Configuration Register (TIMER2_SET_H)

XRAM_SFR Address: 0x206B

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	--	Timer2 count value configuration register, high 8 bits. Configure the register during the scanning process will refresh the counter.

7.3.2.3 Timer2 Configuration Register (TIMER2_SET_L)

XRAM_SFR Address: 0x206C

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	--	Timer2 count value configuration register, low 8 bits. Configure the register during the scanning process will refresh the counter.

The registers TIMER2_SET_H and TIMER2_SET_L are used to configure the time setting.
TTimer2 can be calculated as follows:
TTIMER2=TTIMER2_CLK*({TIMER2_SET_H，TIMER2_SET_L}+1)

7.3.2.4 RTC Configuration Register (RTC_CFG)

XRAM_SFR Address: 0x206D

Bit No.	7~5	6~4	3~2	1	0
Symbol	--	RTC_CLK_FRE	RTC_CLK_SEL	RTC_RLD	RTC_EN
Read/Write	--	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	--	0	0	0	0

Bit No.	Symbol	Description
6~4	RTC_CLK_FRE	RTC module XTAL or PLL_24KHz clock frequency division selection. 000: selects clock frequency/1 001: selects clock frequency/2 010: selects clock frequency/4 011: selects clock frequency/8 100: selects clock frequency/16 101: selects clock frequency/32 110: selects clock frequency/64 111: selects clock frequency/128 No matter in which mode, configure the register during the counting will clear the counter. If the present counter enable is valid, it will count from 0 again.
3~2	RTC_CLK_SEL	RTC clock selection register. 00: select RC32KHz 01: select XTAL 10/11: select PLL_24MHz clock frequency division
1	RTC_RLD	RTC auto-reload enable register. 1: auto-reload mode. 0: manual-reload mode.
0	RTC_EN	RTC counter enable register. Configure 1 to enable the clock. Configure 0 to stop it. In manual reload mode, hardware will auto-clear the register after the counting and stop the clock. In auto-reload mode, the register holds the enable after the counting, and clock starts to count from 0 again. In either mode, configure the register as 1 will enable the clock to count from 0.

Configure the RTC_CLK_SEL/RTC_CLK_FRE to select RTC counter operation.

Note:

- Check if the clock is enabled.
- PD_XTAL=0x1, disable XTAL clock. PD_XTAL=0x0, enable XTAL. By default, this clock is off.
- In single-cycle timing mode, when the timing is up, it transmits an interrupt signal, then the interrupt falling edge will auto-clear the register.
- In auto-reload mode, when the timing is completed, interrupt signal is sent, enable register holding, the clock goes to the next round of timing.
- Configure the register at any time will reset it, therefore, after the interrupt signal is sent, the register will auto-clear the counter and the flag. If the configuration is 1 at this time, it will count from 0.
- In auto-reload mode, concerning the accuracy, it may not to configure the timer registers in the interrupt handler functions, because such operations will clear the counter.

7.3.2.5 RTC Configuration Register (RTC_SET_H)

XRAM_SFR Address: 0x206E

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	--	RTC count value configuration register, high 8 bits. Configure the register during the scanning process will refresh the counter.

7.3.2.6 RTC Configuration Register (RTC_SET_L)

XRAM_SFR Address: 0x206F

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description

7~0	--	RTC count value configuration register, low 8 bits. Configure the register during the scanning process will refresh the counter.

RTC _SET_H and RTC _SET_L are used for time setting.
TRTC can be calculated as follows:
TRTC=TRTC_CLK*({RTC_SET_H, RTC_SET_L}+1)

7.4 Timer Configuration Flow Diagram

7.4.1 Timer0/1 Configuration Flow

Timer0/1 configuration flow as follows:

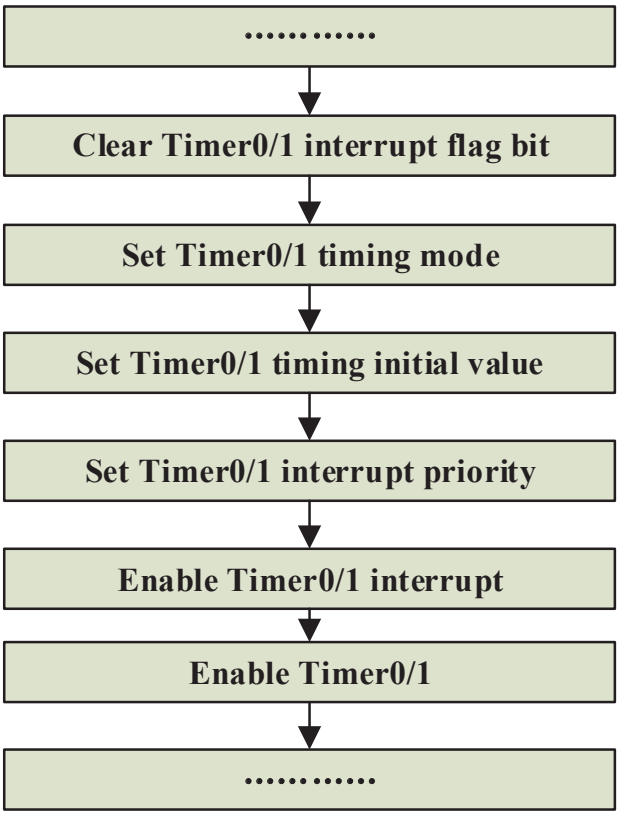


Figure 0.5 Timer0/1 Configuration

In the configuration process of Timer0/1,the initial timer value of Timer0/1 is calculated according to different timing modes.

- 16bit mode.TH0=(65536-N) /256， TL0=(65536-N)%256， N=t/Tclk
- 13bit mode.TH0=(8192-N) /32， TL0=(8192-N)%32， N=t/Tclk
- 8bit mode TH0=(256-N)， TL0=(256-N)， N=t/Tclk。

where t is the timing time set according to the demand,Tclk is the system clock F_sys_clk.

7.4.2 Timer2/RTC Configuration Flow Diagram

Configure TIMER2_CLK_SEL/TIMER2_CLK_FRE to select the Timer2 clock.
PD_XTAL=0x1, disable XTAL. PD_XTAL=0x0, enable XTAL. By default, it is off. Check if the selected clock is enabled. RTC has the same configuration as the Timer2.

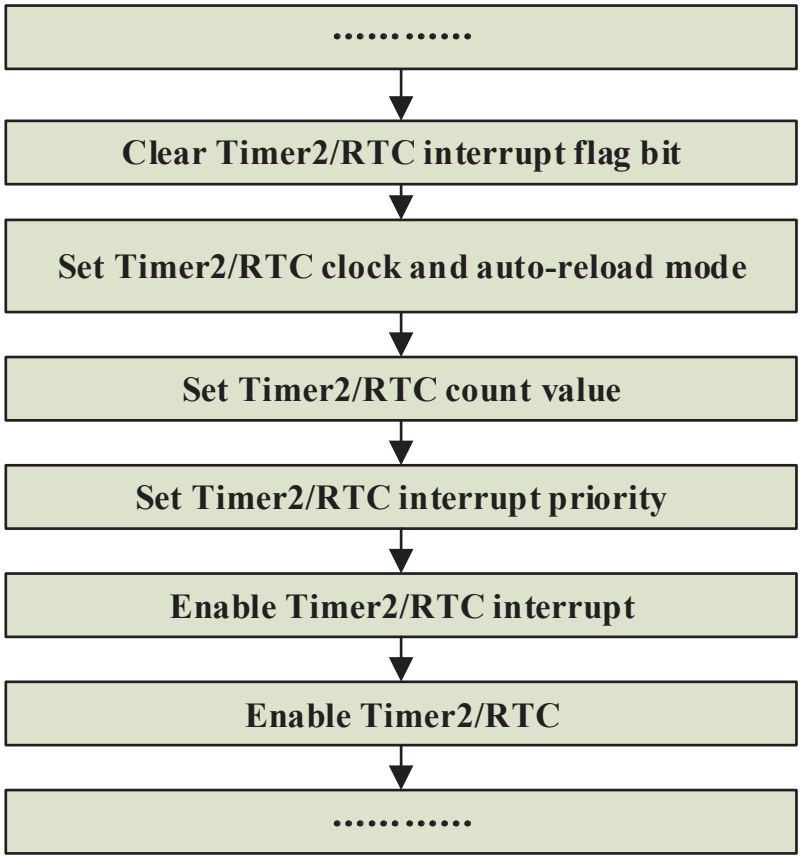


Figure0.6 Timer2/RTC configuration

Timer2/RTC Configuration Steps:

1. First, to set the frequency and clock source, configure {TIMER2_CLK_SEL/TIMER2_CLK_FRE}/{RTC_CLK_SEL/RTC_CLK_FRE}.
2. Then, Configure auto-reload enable of TIMER2_RLD/RTC_RLD accordingly. Set to 1 for auto cycle counting, otherwise set to 0.
3. Finally, configure the TIMER2_EN/RTC_EN. Set {TIMER2_EN=1}/{RTC_EN=1} to enable the counting.
4. Stop the counting. {TIMER2_EN=0}/{RTC_EN=0}.

Note:

- Put the configuration of {TIMER2_EN=0x1}/{RTC_EN=0x1} to the end.
- During the Timer2/RTC running, it is not recommended to change the configurations. If change is necessary, stop the timing first.
- If accuracy is concerned in auto-reload mode, it is not allowed to configure Timer2/RTC and reload the mode during the interrupt handling.

8 PWM

Timer Functions Comparison

Timer	Counter width	Type	Prescaler factor	Capture/ Compare Channel	Complementary outputs	Repetition counter	External trigger input	External break input	Timer Synchronous cascade
PWM0 Advanced Timer	16-bit	Up/ Down	1 to 65536	4	3	Yes	1	1	With PWM1
PWM1 General purpose Timer		Up	2^n n=1 to 15	3	No	No	0	0	Yes

Figure 0.1 Timers Comparison

Timer Signal Terminology

Internal Signal Name	Description	Related Diagram
BI	Break Interrupt	*Figure 8.3 PWM0 Diagram*
CCxI, CC1I, CC2I, CC3I, CC4I	Capture/Compare Interrupt	
CK_CNT	Counter	*Figure 8.8 Prescaler factor 2 Counter without Preload*
CK_PSC	Prescaler	
CNT_EN	Counter Enable	
CNT_INIT	Counter initiation	*Figure 0.20 External Clock Source Mode 1 Block* Diagram
ETR	TIMx_ETR signal	*Figure 8.22 External Trigger Input Diagram*
ETRF	External Trigger Filter	Figure 0.22 External Trigger Input Block Diagram
ETRP	External Trigger Prescaler	
f_{MASTER}	Peripheral Clock from Clock Controller (CLK)	*Figure 1.2 Clock Diagram*
ICx, IC1, IC2	Input Capture	*Figure 8.41 PWM0 Channel 1 Input*

ICxPS, IC1PS, IC2PS	Input Capture Prescaler	
MATCH1	Compare Match	*Figure 8.32 Mster/Slave Mode Connection Diagram*
OCx, OC1, OC2	Timer Output Channel	*Figure 8.46 Detailed Output Module Diagram（channel 1）with Complementary Output*
OCxREF, OC1REF, OC2REF	Output Compare Reference Signal	
TGI	Trigger Interrupt	
TIx, TI1, TI2	Timer Input	*Figure 8.41 PWM0 Channel 1 Input*
TIxF, TI1F, TI2F	Timer Input Filter	
TI1_ED	Timer Input edge detection	
TIxFPx, TI1FP1, TI1FP2, TI2FP1, TI2FP2	Timer Input Filter Prescaler	
TRC	Trigger Capture	
TRGI	Clock/Trigger/Slave Mode Controller Trigger Input	*Figure 8.19 Control circuit sequence diagram in Normal Mode with f_{MASTER} Prescaler Factor 1*
UEV	Update Event	*Figure 8.12 Figure 0.12 Non-preload Counter with Prescaler Factor of* 2
UIF	Update Interrupt	

Table 0.2 Internal Timer Signal Terminology

8.1 16-bit Advanced Timer PWM0

PWM0 consists of a 16-bit up/down auto-reload counter and a programmable prescaler that serves as a driver. In this chapter, x represents 1, 2, 3, 4, corresponding to 4 different capture/compare channels, respectively. Advanced control timer is suitable for various applications.

1. Basic Timing.
2. Measure pulse width of input signal (Input Capture)
3. generate output waveform (Output Compare, PWM and One Pulse Mode)
4. Correspond to interrupts of different events (capture, compare, overflow, break, trigger).
5. Sync with PWM1 or External Signals (external clock, reset signal, trigger and enable signal.

Advanced timers are widely used in all kinds of control applications, including those PWM requiring

center-aligned mode. This mode supports complementary output and dead-time control.
The clock source of advanced timer can be PLL_24MHz, or external oscillator XTAL, which can be configured the register to select.

8.1.1 Features

PWM0:

- 16-bit up, down, up/down auto-reload counter.
- Repetition counter allowing to update the timer register after the designated cycles
- 16-bit programmable (support real-time modification) prescaler PWM0_PSC, the prescaler factor is within 1 to 65536.
- Synchronous circuit used in external signal control timer and timer interconnection.
- 4 independent channels support the configuration as follows:
  - Input Capture
  - Output Compare
  - PWM Generation (edge-aligned or center-aligned)
  - Six steps for PWM output
  - Output in one pulse mode
  - 3 dead-time programmable channels with complementary output and
- Break input signal can put timer output signal into reset state or a known state.
- Events that triggers interrupt are including:
  - Update: counter overflow/underflow, counter initiation (via software or internal/external trigger)
  - Trigger events (counter startup, stop, initiation or internal/external trigger)
  - Input capture
  - Output compare
  - Break signal input

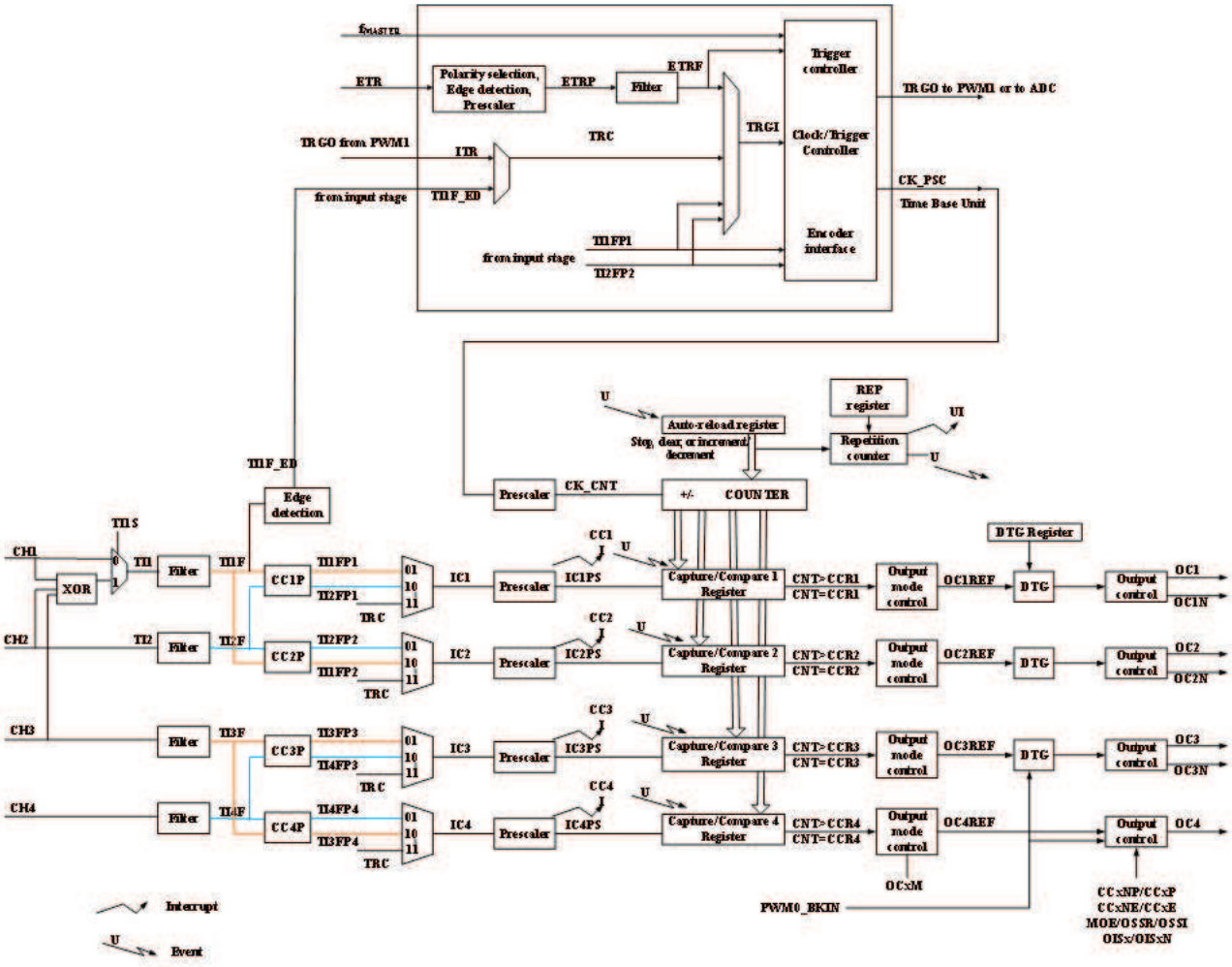


Figure 0.3 PWM0 Diagram

8.1.2 Time Base Unit

Time Base Unit includes a 16-bit up/down counter, a prescaler, a 16-bit auto-reload register and repetition counter register. The read/write can be configured in software.

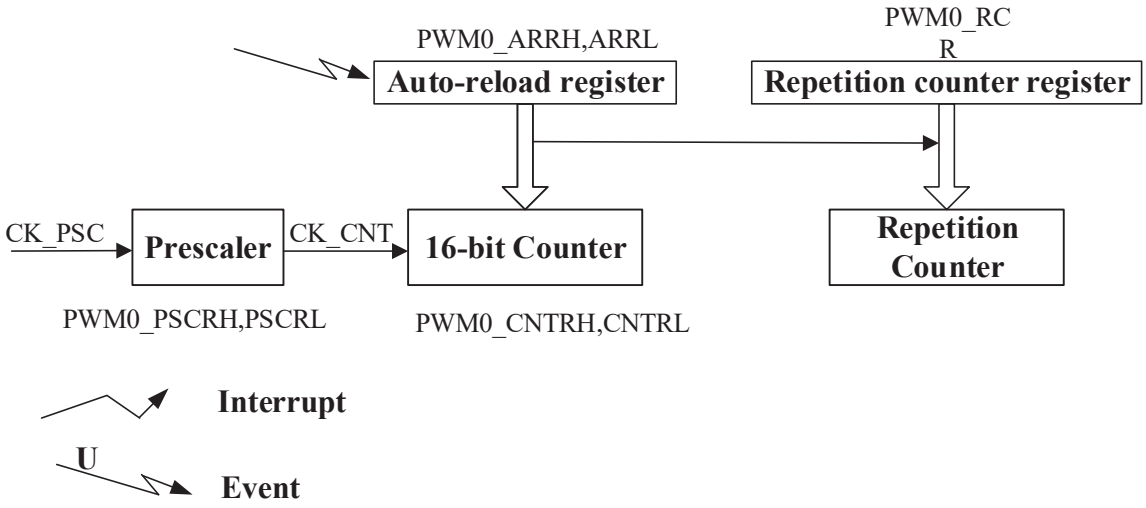


Figure 0.4 Time Base Unit

Auto-reload register consists of preload registers and shadow registers, supporting the write auto-reload register in two modes below:

1. **Auto-preload enabled** (set the ARPE bit in the PWM0_CR1), data written into auto reload register will be stored in preload register and passed to shadow register when the next update event happens.
2. **Auto-preload disabled** (clear the ARPE bit in the PWM0_CR1), data written into auto reload register will immediately be written into shadow register.

Update events may generate three conditions as follows:

1. Counter overflow or underflow.
2. Set the UG bit in the PWM0_EGR.
3. Clock/trigger controller generates a trigger event.

When preload is enabled (ARPE=1), if an update event occurs, data in preload register (PWM0_ARR) will be written into shadow register, and data in PWM0_PSCR register will be written to prescaler.
Set the UDIS bit in PWM0_CR1 will disable update events (UEV).
Counter is driven by CK_CNT of prescaler, however, CK_CNT is only valid when CEN bit in PWM0_CR1 is enabled.

Note:
After the CEN bit is enabled for a cycle, the counter starts to count.

8.1.2.1 Read/Write 16-bit Counter

Write operation on counter is not cached and can write registers PWM0_CNTRH and PWM0_CNTRL at any time. Therefore, it is not recommended to write a new value in it when counter is running, in case any wrong value is written into it.
Read operation on counter has an 8-bit cache. After user reads high byte, low byte will be automatically cached. Cached data will not change before the16-bit read operation is completed.

(See figure 8.5)

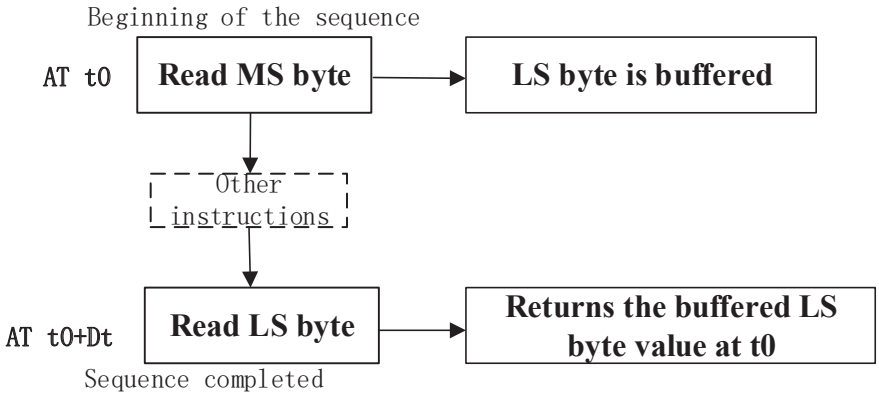


Figure 0.5 Read a 16-bit counter (PWM0_CNTR)

Note: Do not read out low byte first, because the read-out value is wrong.

8.1.2.2 Read/Write 16-bit register PWM0_ARR

Write the value of Preload register into the 16-bit register PWM0_ARR with high byte first and low byte next. Shadow register will be locked during the high byte write operation and holding the state till the low byte write operation is competed.
Note: Do not write low byte first, which will result in data error.
To read out the PWM0_ARR requires to preload the register.

8.1.2.3 Prescaler

PWM0 prescaler is based on a 16-bit counter controlled by a 16-bit register (PWM0_PSCR). Since the control register has a buffer, therefore, it can be changed during the running. Prescaler can divide the clock frequency of counter by any integer from 1 to 65536. Counter frequency can be calculated as follows:

$$f_{CK\_CNT} = f_{CK\_psc}/(PSCR[15:0] + 1)$$

Write the value of prescaler via preload register. Shadow register contained the present value is loaded when LSB is written into.
Prescaler value requires two independent write operations to write the 16-bit register. Write high bits first and do not use LDW command to write low bits.
The new value of prescaler will be adopted for the next update event.
Read the PWM0_PSCR through preload register, which requires no special attention.

8.1.2.4 Up-counting Mode

In up-counting mode, counter first counts from 0 to the compare value (value of PWM0_ARR register) that defined by user, then restart to count and generate an overflow event. Concurrently, if UDIS bit of PWM0_CR1 is 0, it will generate an update event (UEV) that will update the flag bit UIF. If update

interrupt enables UIE=1, it generates an update interrupt. Figure 8.6 shows the up-counting mode.

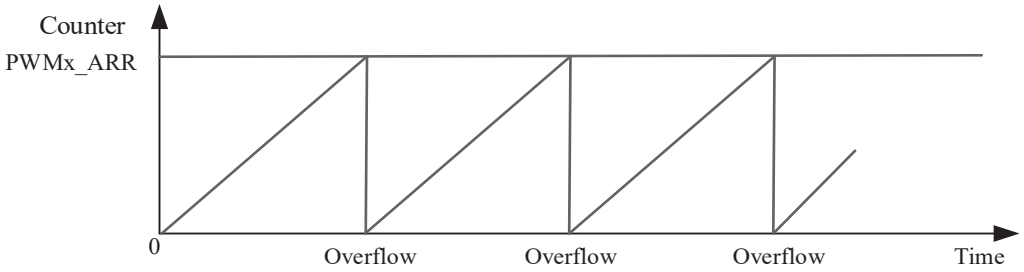


Figure 0.6 Up Counter

Set the UG bit in PWM0_EGR (via software or slave controller) can also generate an update event. Use software to set UDIS bit in PWM0_CR1 to ban update events, which can avoid updating shadow register while updating the preload register. No update event will be generated before the UDIS bit is cleared. Although when there should be an update event, the counter shall still be cleared to 0, concurrently prescaler will be cleared too (its value remains unchanged). Moreover, if the URS bit is set in the PWM0_CR1 (select update request). Set UG bit will generate an update event. However, hardware does not set UIF flag, which means it does not generate an interrupt request. This is to avoid generating an update and capture interrupt while clearing the counter in capture mode.

When an update event occurs, all registers will be updated. Concurrently, hardware sets the update flag bit (UIF bit in the PWM0_SR) according to the URS bit.

- auto-reload the value of shadow register that reset into preload register (PWM0_ARR).
- Set the value of preload register PWM0_PSC into the buffer of prescaler.

Figure 8.7 shows the update sequence of an up counter when ARPE=1(ARR preload enable).

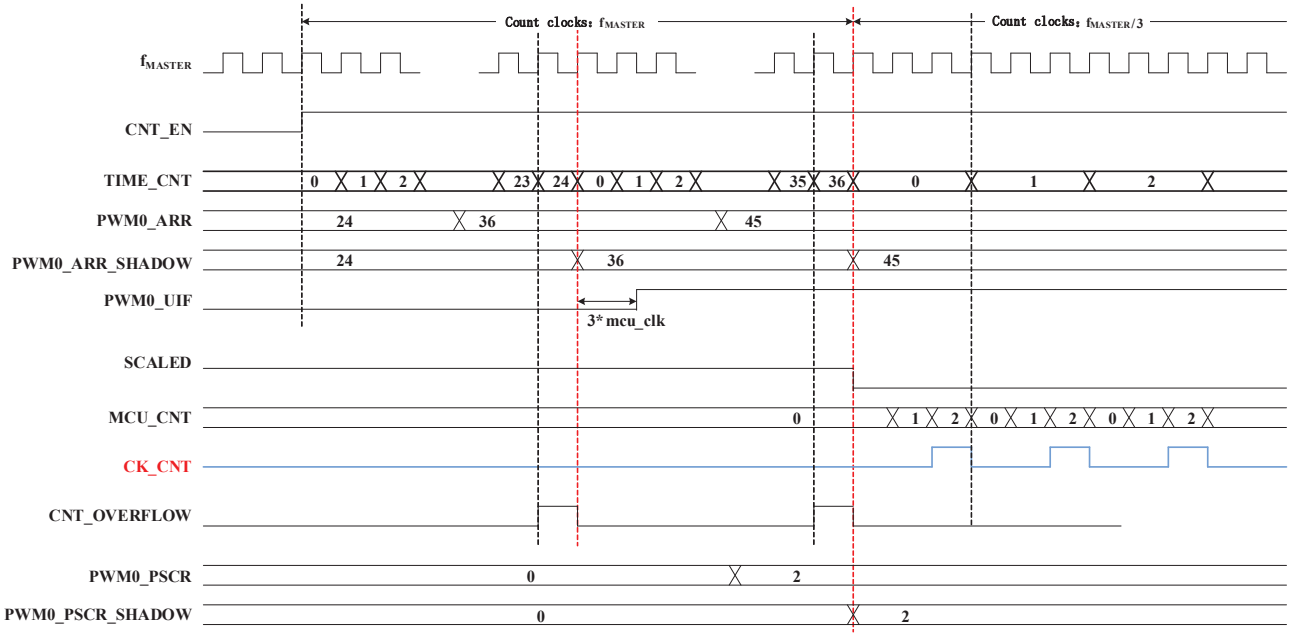


Figure 0.7 Up Counter Sequence Diagram (In PWM0_CR1, ARPE=1)

Figure 8.8 shows an example of counter behavior at different clock frequencies when PWM0_ARR=0x36.

Example 1

When ARPE=0 (ARR without preload), the counter be updated with prescaler factor of 2 .

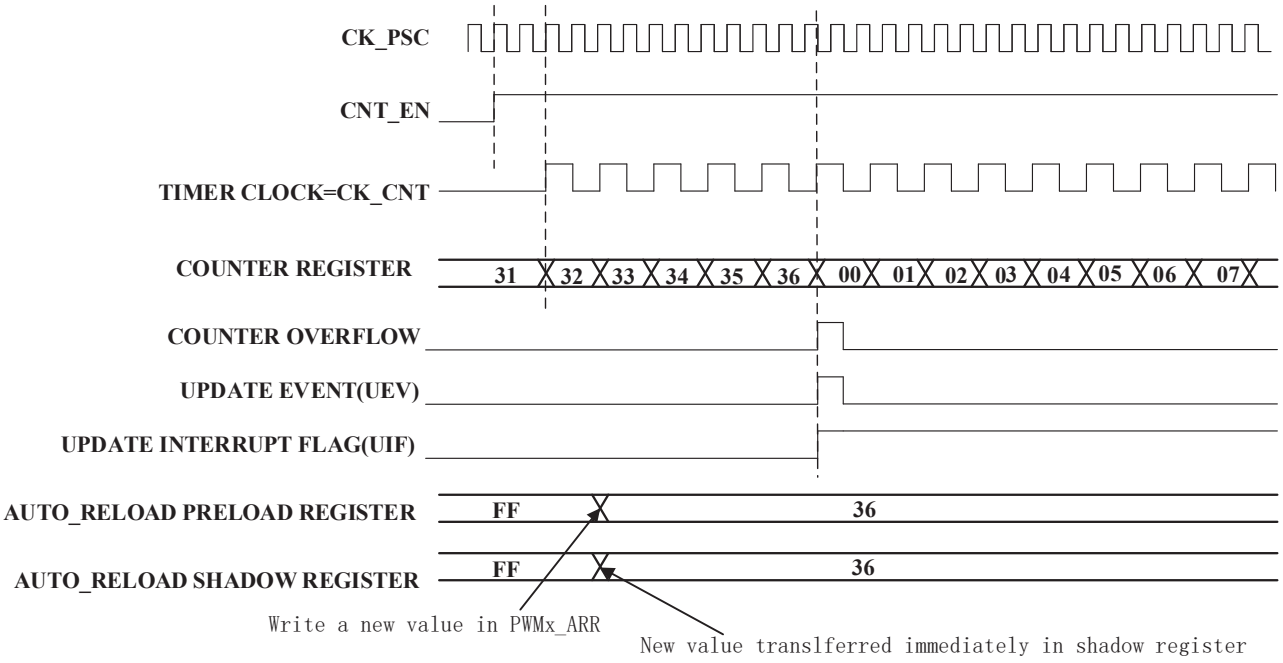


Figure 0.8 Non-preload Counter with Prescaler Factor of 2

Since the prescaler factor is 2, the counter (CK_CNT) is half the frequency of prescaler (CK_PSC). The counter overflows when reaching to 0x36 because the auto-reload function is disabled (ARPE=0), then the shadow register will be updated and generates an Update Event.

Example 2

When ARPE=1(ARR preload enable), the counter be updated with prescaler factor of 1.

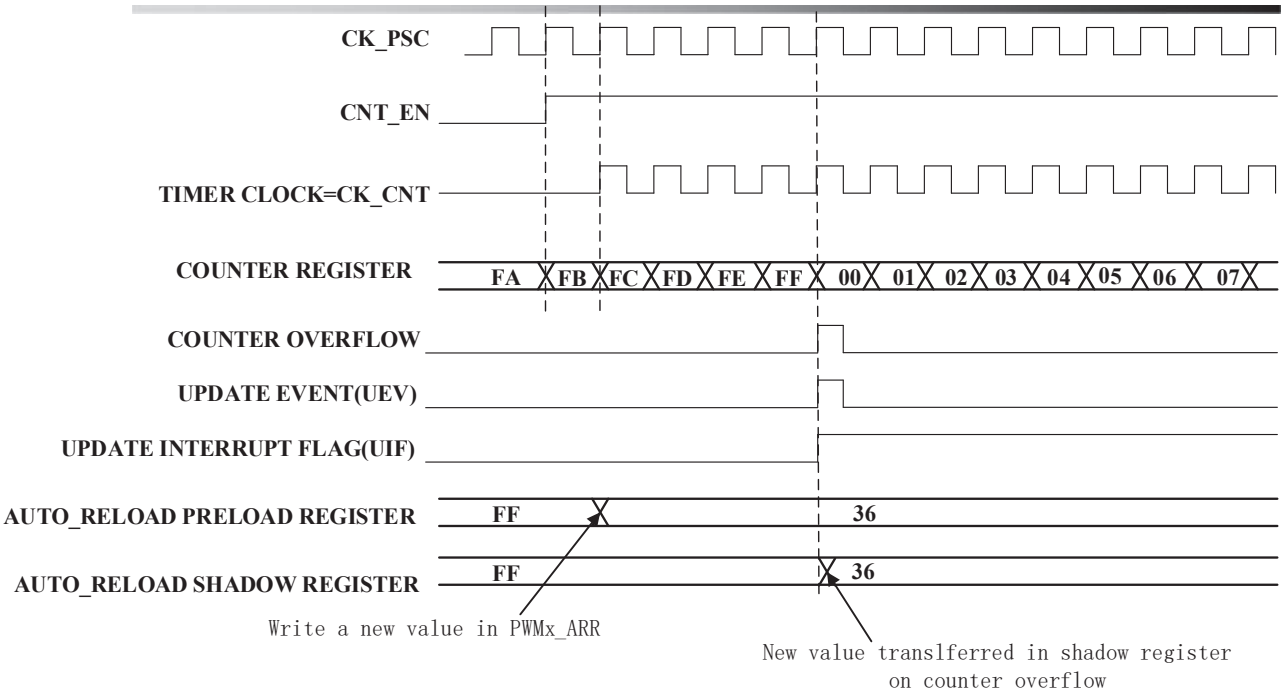


Figure 0.9 Preload Counter with Prescaler Factor of 1

Prescaler Factor is 1, therefore the frequency of CK_CNT is the same with CK_PSC.
Enable auto-reload as ARPE=1, therefore counter will overflows when reaching to 0xFF, then 0x36 will be written upon overflow, generating an update event at the same time.

8.1.2.5 Down-counting Mode

In down-counting mode, counter counts down from the the auto-reload value (contained in PWM0_ARR) to 0, then starts to count from the auto-reload value again and generate an underflow event. If UDIS bit in PWM0_CR1 is cleared, it will generate an UEV,which will update the flag bit UIF. If update interrupt enable is set UIE=1, it will trigger an update interrupt. Figure 8.10 demonstrates the down counter.

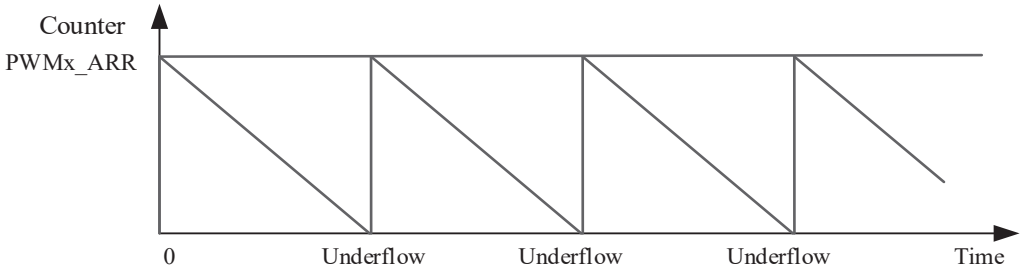


Figure 0.10 Down Counter

Set the UG bit in PWM0_EGR via software or slave controller can also generate an UEV.
Use software to set UDIS bit in PWM0_CR1 can banUEVs, avoiding to update the shadow register while updating the preload register. No UEV occurs before the UDIS bit is cleared. Nevertheless,

counter still restarts to count from the present auto-reload value, and the counter of prescaler begins to count from 0 again (prescaler cannot be modified). Moreover, if set URS bit in PWM0_CR1 to select update request, setting UG bit will generate an UEV. However, hardware does not set UIF bit, which means it does not trigger interrupt request. This is to avoid generating an update and capture interrupt while clearing the counter in capture mode.

When an UEV occurs, all registers will be updated and reset the update flag bit (UIF in PWM0_SR) according to URS bit setting.

- Reset the value of preload register (PWM0_ARR) into auto-reload shadow register.
- Reset the value of PWM0_PSC into Buffer of prescaler.

The present auto-reload register is updated into the preload value in PWM0_ARR. It is noted that auto-reload register is updated before the counter reloads, therefore, the expected value is in the next cycle.

Figure 8.11 shows the down counter update sequence diagram when ARPE=1(ARR preload enable).

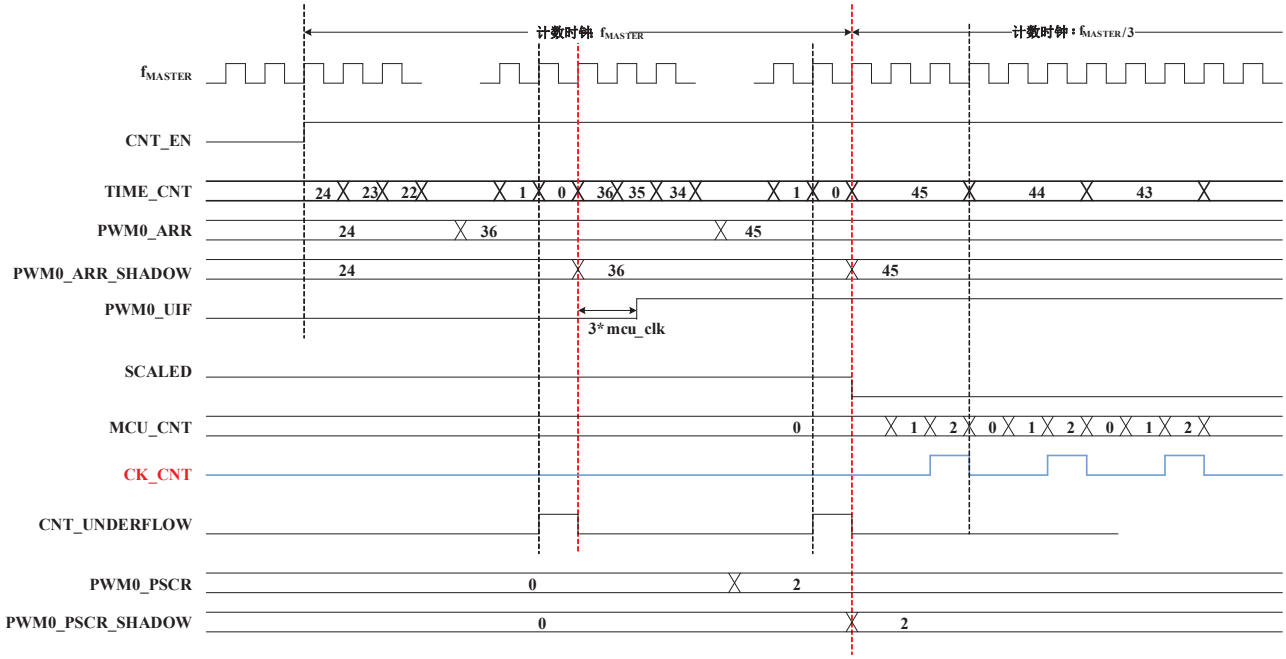


Figure 0.11 Down Counter Sequence Diagram (In PWM0_CR1, ARPE=1)

Figure 8.12 shows an example of counter behavior at different clock frequencies when PWM0_ARR=0x36.

Example 1

When ARPE=0 (ARR without preload), the counter be updated with prescaler factor of 2.

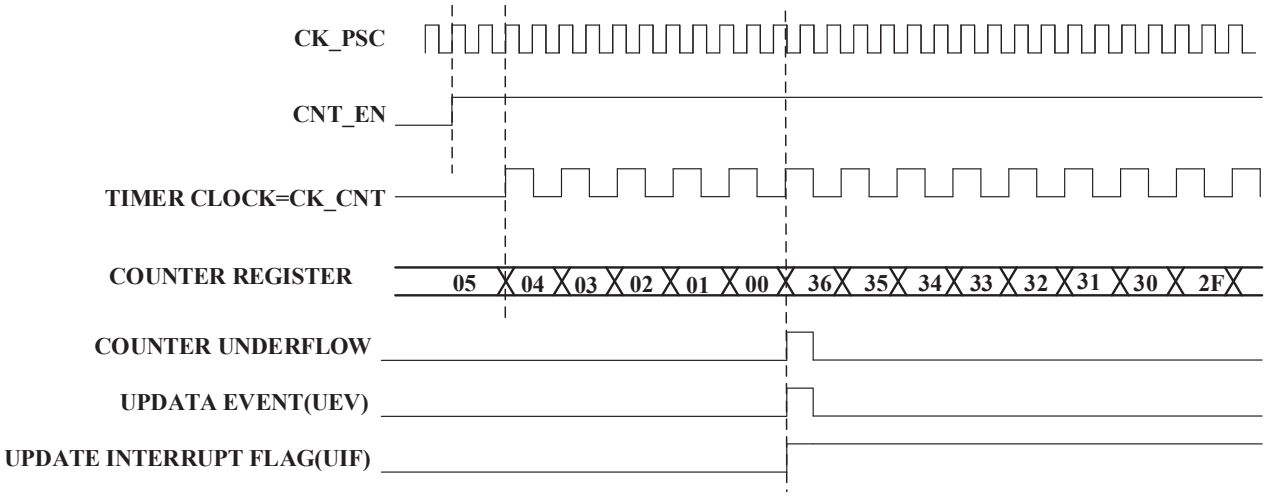


Figure 0.12 Non-preload Counter with Prescaler Factor of 2

Example 2

When ARPE=1(ARR preload enable), counter with prescaler factor of 1 will be updated.

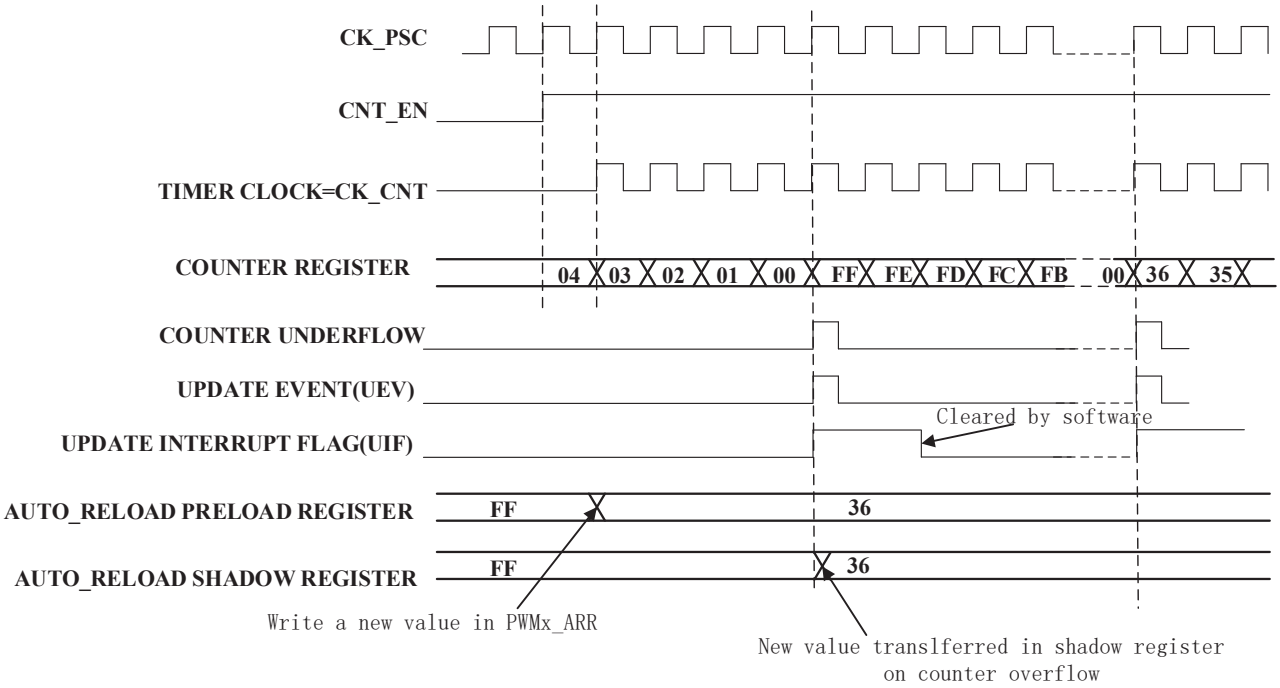


Figure 0.13 Preload Counter with Prescaler Factor of 1

8.1.2.6 Center Aligned Mode (Count Up/Down)

In center aligned mode, counter counts up from 0 to the auto-reload value (PWM0_ARR) minus 1, which generates an overflow event. Then, it counts down from auto-reload value to 0 and generates an underflow event. After that, it counts from 0 again.

In this mode,direction bit DIR of PWM0_CR1 can not be written, which is updated by hardware and

points to the present counting direction.

Figure 8.14 shows the center aligned mode.

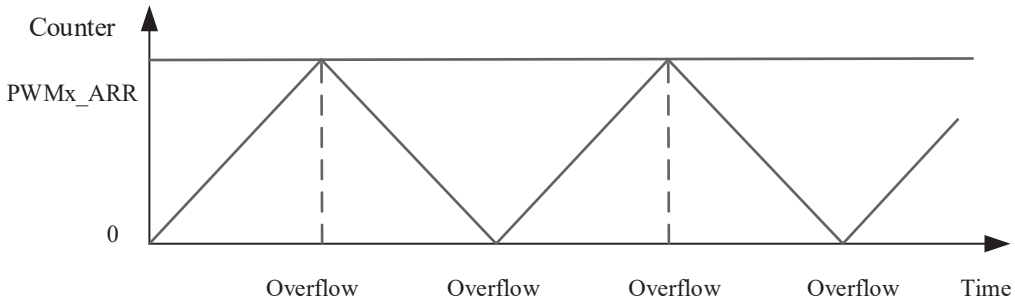


Figure 0.14 Counter in Center Aligned Mode

If the repetition counter of timer does not equal to 0, then it will generate an UEV after repeating the designated times (value of PWM0_RCR) of overflow and underflow. Otherwise, for every overflow or underflow, an update event will be generated.

Set the UG bit in PWM0_EGR via software or slave controller can also generate an update event. At this time, counter initializes and restarts to count, and prescaler also initializes and restarts to count.

Set the UDIS bit to 1 in PWM0_CR1 to ban UEV, which avoids updating shadow register when updating the preload register. Therefore, no update event occurs before the UDIS bit is cleared to 0. However, counter still counts up or down according to auto-reload value. For timer with a repetition counter, new repeat value will be immediately valid because there is no double buffer in repetition register, so be careful when modifying it.

Moreover, if set the URS bit in PWM0_CR1 (select update request), then setting the UG bit will generate an UEV. However, hardware does not set UIF flag bit, which means it does not trigger interrupt request. This is to avoid generating an update and capture interrupt while clearing the counter in capture mode.

When an UEV occurs, all registers will be updated and set the update flag bit (UIF in PWM0_SR) according to URS bit setting.

- If the ARPE bit of PWM0_CR1 is set to 1, auto-reload shadow register will be set with the value of preload register (PWM0_ARR).
- Prescaler buffer is set with the value of preload register (PWM0_PSCR).

The present auto-reload register is updated to the preload value of PWMx_ARR. It is noted that if it is counter overflow that generates the update, auto-reload register is updated before the counter reloads, so the expected value is in the next cycle, which means counter will be loaded with new value.

Figure 8.15 shows the center aligned counter update sequence when ARPE=1 (ARR preload enabled).

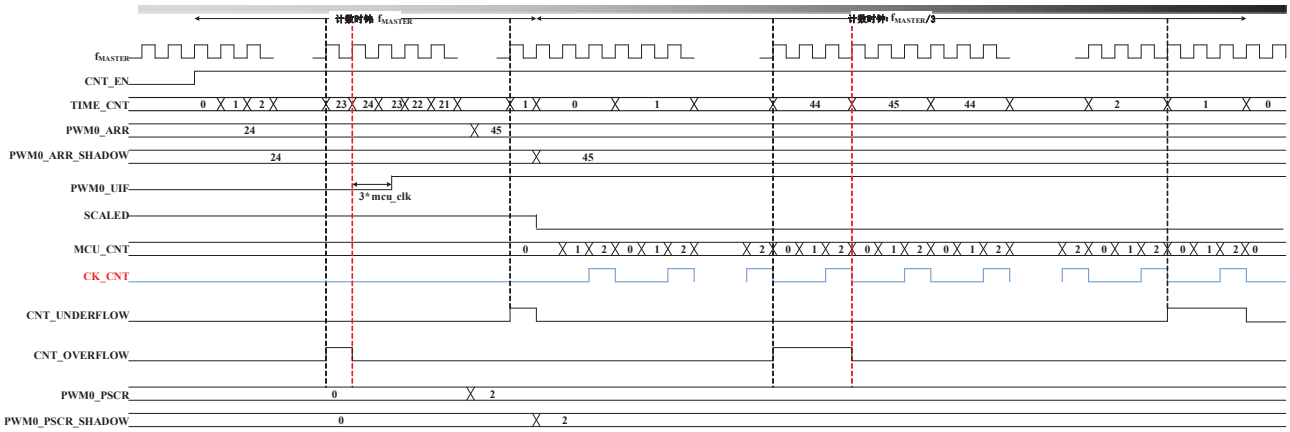


Figure 0.15 Center Aligned Counter Sequence Diagram (In PWM0_CR1, enable ARPE=1)

Figure 8.16 shows the counter sequence diagram when internal clock with a prescaler factor of 1 PWM0_ARR=0x6, ARPE＝1.

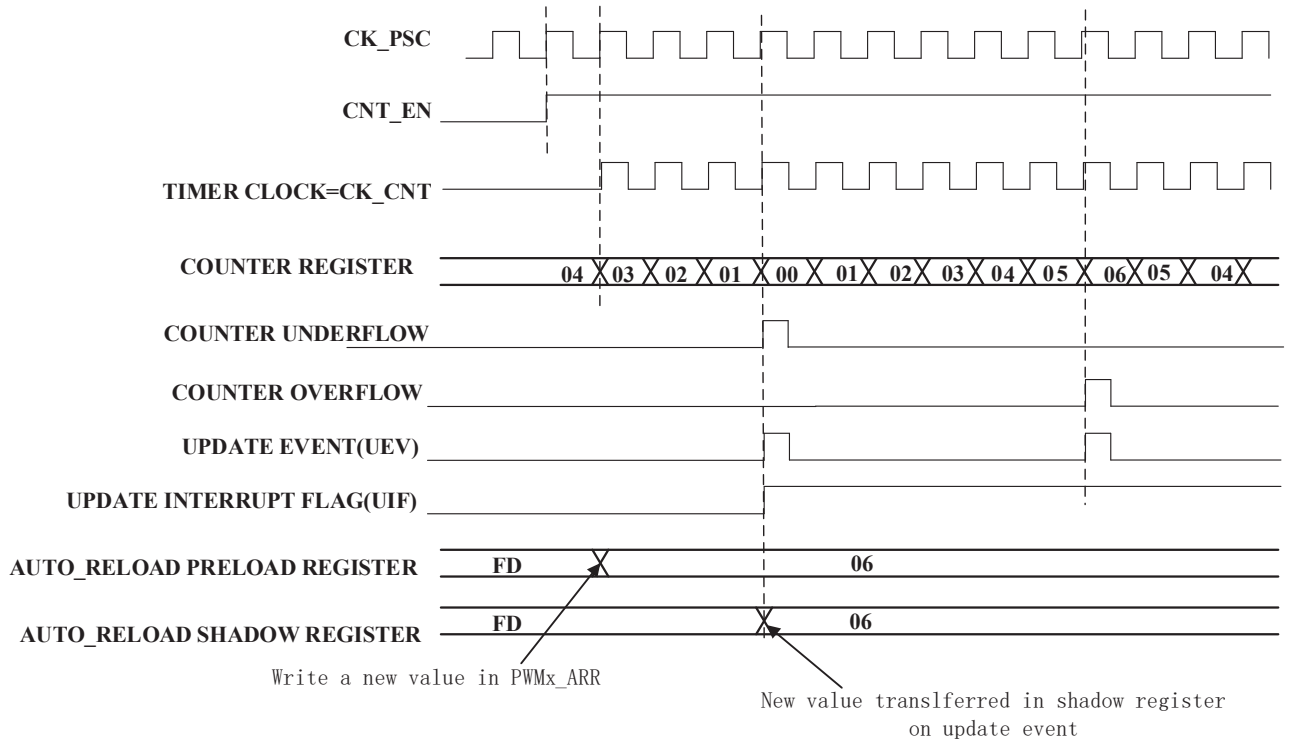


Figure 0.16 Preload Counter Sequence Diagram with Prescaler Factor of 1

Notes on center aligned mode:

- Enabled center aligned mode, counter will count according to the previous up/down configuration, which means the DIR bit of PWM0_CR1 will decide to count up or count down. Software cannot modify the DIR and CMS bit at the same time.
- It is not recommended to write on counter during the counting in this mode, which will result

in unexpected results as follows:

- Down counter writes a value larger than auto-reload value to counter (PWM0_CNT> PWM0_ARR), but its counting direction will not change. For example, counter has overflowed, but it still counts up.
- Down counter writes into 0 or value of PWM0_ARR, but no UEV occurs.

- The safe way to use center aligned counter is to use software to generate an UEV before the counter enabled (set the UG bit of PWM0_EGR) and do not change its value during the counting.
- After a cycle (CK_CNT) of *CEN* bit is enabled, the counter begins to count.
- After the counter is enabled, it is not allowed to change edge aligned mode to center aligned mode.
- In this mode, encoder mode (in PWM0_SMCR, SMS=001, 010, 011) must be banned.
- If initial counting direction is configured as down in center aligned mode, configure DIR =1 in PWM0_CR1 first, and configure CMS after delays.

8.1.2.7 Repetition Down Counter

The repeat counter is a counter that counts down. This is because the update event (UEV) can only be generated when the value of the repeat counter reaches 0, which means that every N times of counting overflow or underflow, data is transferred from the preload registers to the shadow registers (auto-reload register PWM0_ARR, preload register PWM0_PSC, as well as the capture/compare register PWM0_CCRx in compare mode), and N is the value in the repeat counter register PWM0_RCR. The repetition counter decrements when either of the following conditions holds:

- Each time the counter overflows in up-counting mode
- Each time the counter underflows in Down-counting mode
- Each time the counter overflows or underflows in center aligned mode

Although this limits the maximum repetition times of PWM to 128, it is able to update the duty cycle 2 times per PWM cycle. In central aligned mode, the waveform should be symmetrical and the maximum resolution is 2xTck_psc if the compare register is refreshed only once in each PWM cycle. The repetition counter is automatically loaded and the repeat rate is defined by the value of PWM0_RCR register, which can be referred to Figure 8.17 below.

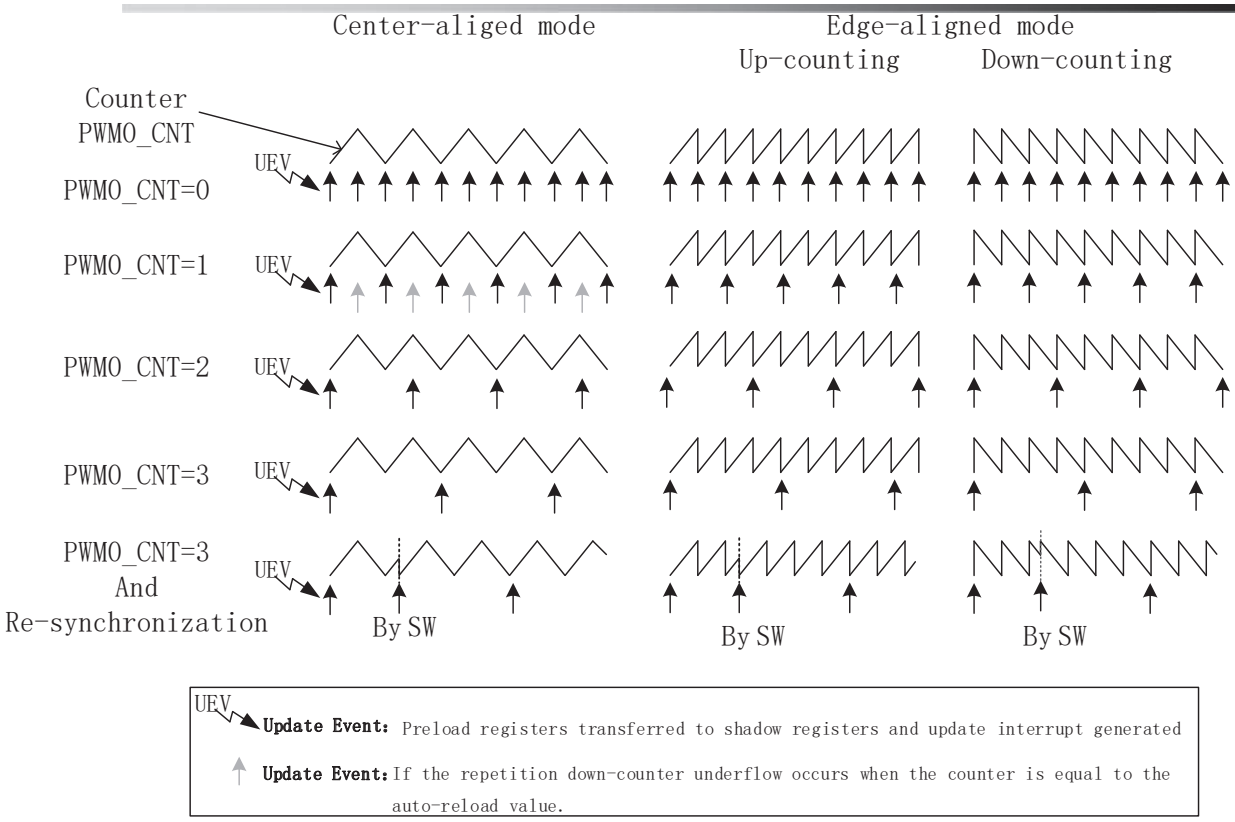


Figure 0.17 Example of update rate in different modes and PWMx_RCR setting

When an update event is generated by software (by setting the UG bit in PWM0_EGR) or by the hardware slave controller, the update event occurs immediately regardless of the repetition counter value, and the contents of the PWM0_RCR register are reloaded into the repetition counter.

8.1.3 Clock/Trigger Controller

The clock/trigger controller allows the user to select the clock source, input trigger signal and output signal for counter.

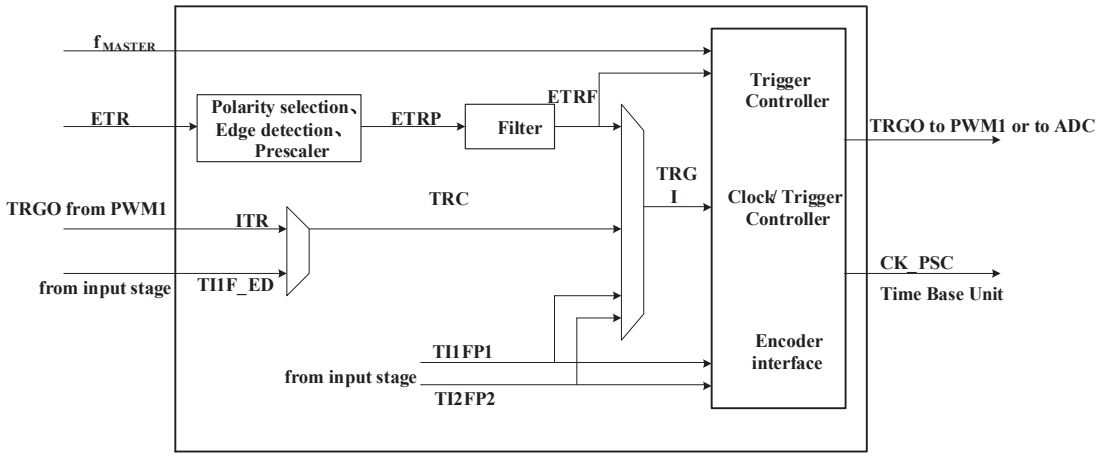


Figure 0.18 Clock/Trigger Controller

8.1.3.1 Prescaler Clock (CK_PSC)

The prescaler (CK_PSC) of the Time Base Unit can be provided by the following sources:

1. Internal clock (f_{MASTER})
2. External clock mode 1: External clock input (TIx)
3. External clock mode 2: External trigger input ETR
4. Internal Trigger Input (ITR): Use the general-purpose timer PWM1 as the prescaler for advanced timer PWM0. For more information, please refer to*Figure 0.32 Master/Slave Connection*for more information.

8.1.3.2 Internal clock source (f_{MASTER})

If both the trigger controller and the external trigger input are disabled (SMS=000 in the PWM0_SMCR register and ECE=0 in the PWM0_ETR register), the CEN, DIR and UG bits are effectively control bits and can only be modified by software (the UG bit is automatically cleared). Once the CEN bit is set to 1, the clock of prescaler is provided by the internal clock f_{MASTER}.
Figure 8.19 demonstrates the operation of the control circuit and up counter in normal mode without prescaler (fMASTER prescaler factor of 1).

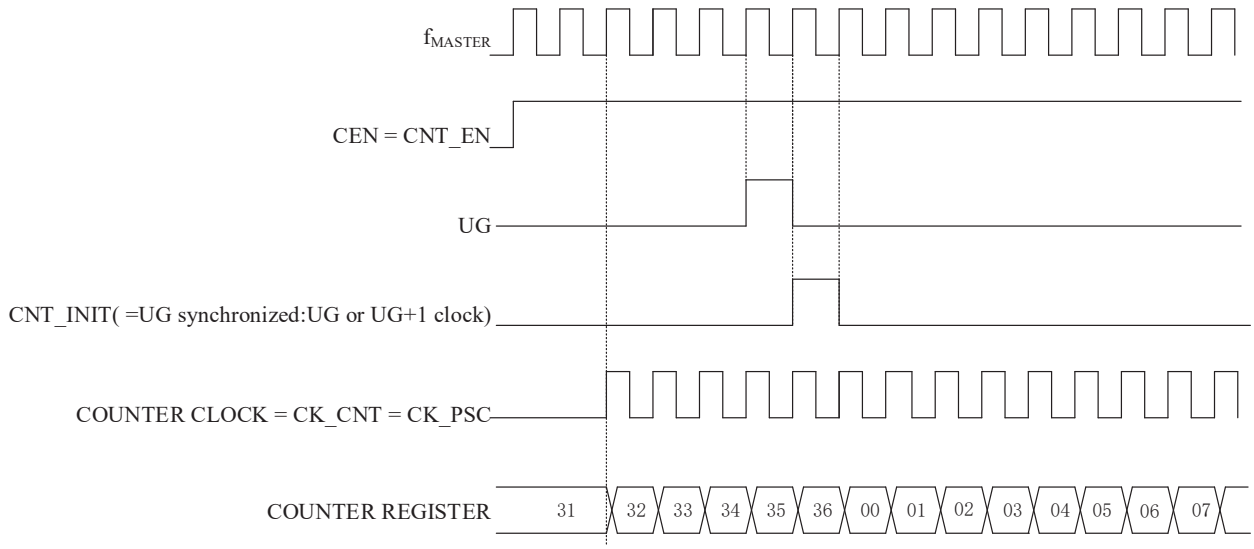


Figure 0.19 Timing diagram of the control circuit in normal mode with f_{MASTER} prescaler factor of 1

8.1.3.3 External Clock Source Mode 1

This mode is selected when SMS=111 in PWM0_SMCR register. The counter can count on each rising or falling edge of the selected input, as shown in Figure 8.20.

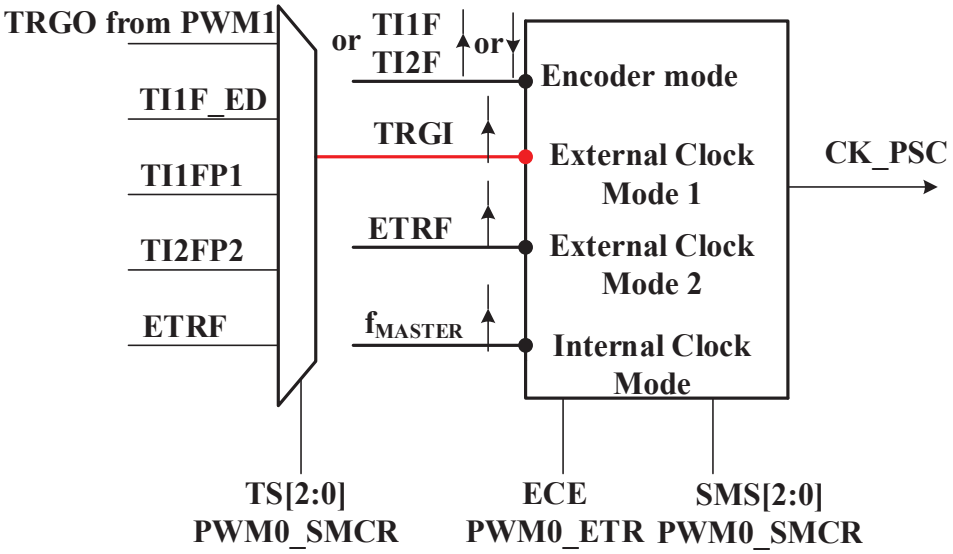


Figure 0.20 External Clock Source Mode 1 Block Diagram

For example, to configure the up counter to count on the rising edge of the TI2 input, use the following steps:

1) Configure the PWM0_CCMR2 register with CC2S=01 and use channel 2 (CC2) to detect the rising edge of TI2 input.
2) Configure the IC2F[3:0] bits of the PWM0_CCMR2 register to select the input filter bandwidth (if no filter is needed, keep IC2F=0000).
   Note: The capture prescaler is not used for triggering and therefore does not need to be configured.
3) Configure the PWM0_CCER1 register with CC2P=0 to select the rising edge polarity.
4) Configure PWM0_SMCR register with SMS=111 to set the counter to use external clock mode 1.
5) Configure the PWM0_SMCR register with TS=110 and select TI2 as the input source.
6) Set CEN=1 of PWM0_CR1 register to start the counter.

When the rising edge occurs at the selected input, the counter counts once and the trigger flag bit (TIF bit in PWM0_SR1 register) is set to 1. If the interrupt is enabled (configured in PWM0_IER register), an interrupt request is generated.

The delay between the rising edge of TI2 and the counter clock depends on the resynchronization circuit at the TI2 input.

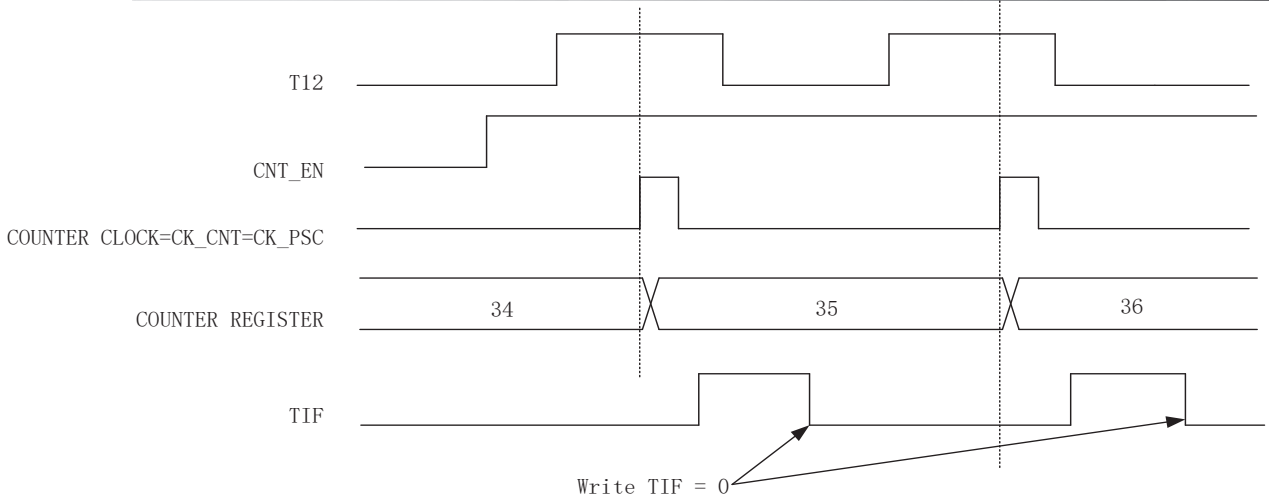


Figure 0.21 Control Circuit in External Clock Mode 1

8.1.3.4 External Clock Source Mode 2

The counter is capable of counting on each rising or falling edge of the External Trigger Input (ETR) signal. This mode can be selected by writing 1 to the ECE bit of the PWM0_ETR register. Figure 8.22 demonstrates the general block diagram of the external trigger input.

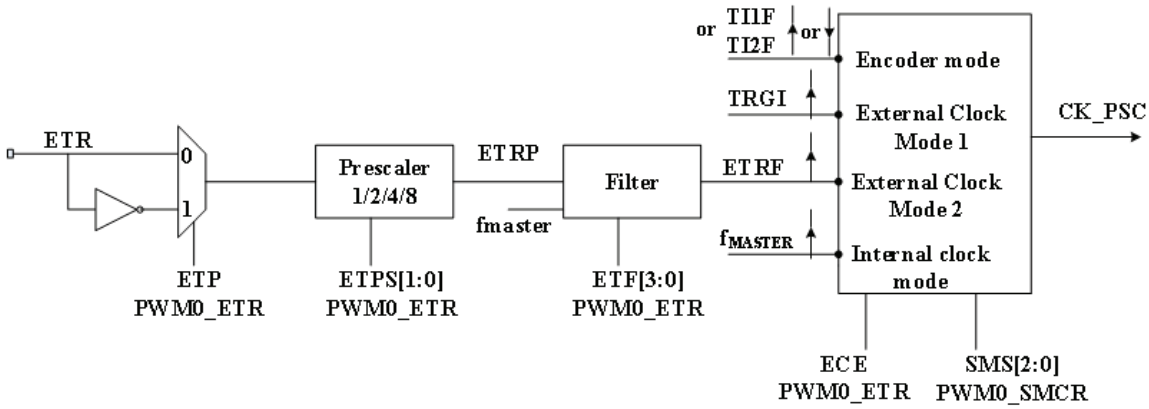


Figure 0.22 External Trigger Input Block Diagram

For example, to configure the counter to count up once every 2 rising edges of the ETR signal, use the following steps:

1) In this example, no filter is needed, configure the PWM0_ETR register with ETF[3:0]=0000
2) Set prescaler, configure ETPS[1:0]=01 of PWM0_ETR register
3) Select the rising edge detection of ETR, configure ETP=0 of PWM0_ETR register
4) Enable external clock mode 2, configure ECE=1 in PWM0_ETR register
5) Start the counter, write CEN=1 in PWM0_CR1 register

6) The counter counts every 2 rising edges of the ETR.
7) The delay between the rising edge of the ETR and the counter clock depends on the resynchronization circuit at the ETRP signal side.

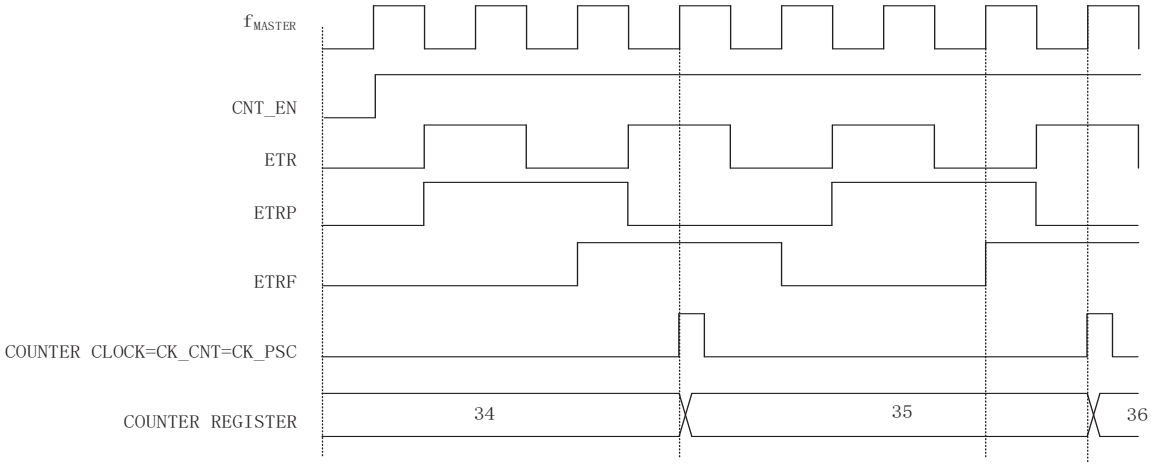


Figure 0.23 Control Circuit in External Clock Mode 2

8.1.3.5 Trigger Synchronization

The counter allows four trigger inputs (Refer to *Table 0.2 Internal Timer Signal Terminology*)

1) ETR
2) TI1
3) TI2
4) TRGO from PWM1

The PWM0 counter uses three modes to synchronize with the external trigger signal: standard trigger mode, reset trigger mode, and gate trigger mode.

When TRGI occurs a rising edge, the counter starts counting driven by the internal clock f_{MASTER} and set TIF flag bit. The delay between the rising edge of the TRGI and the counter beginning counting depends on the resynchronization circuit at the TRGI input.

8.1.3.5.1 Standard Trigger Mode

The counter enable depends on the event from the selected input.

In Figure 8.24, the counter starts counting up on the rising edge of the TI2 input.

- Configure channel 2 (CC2) to detect the rising edge of TI2. Configure the input filter bandwidth (in this example, no filter is needed, keep IC2F=0000). The capture prescaler is not used in trigger operation and does not need to be configured. Configure the PWM0_CCER1 register with CC2P = 0 and select the rising edge as the trigger condition.
- Configure SMS=110 of PWM0_SMCR register to select counter as trigger mode; configure

TS=110 in PWM0_SMCR register to select TI2 as input source.

When a rising edge appears on TI2, the counter starts counting driven by the internal clock and the TIF flag is set at the same time.

The delay between the rising edge of TI2 and the start of the counter depends on the resynchronization circuit at the TI2 input.

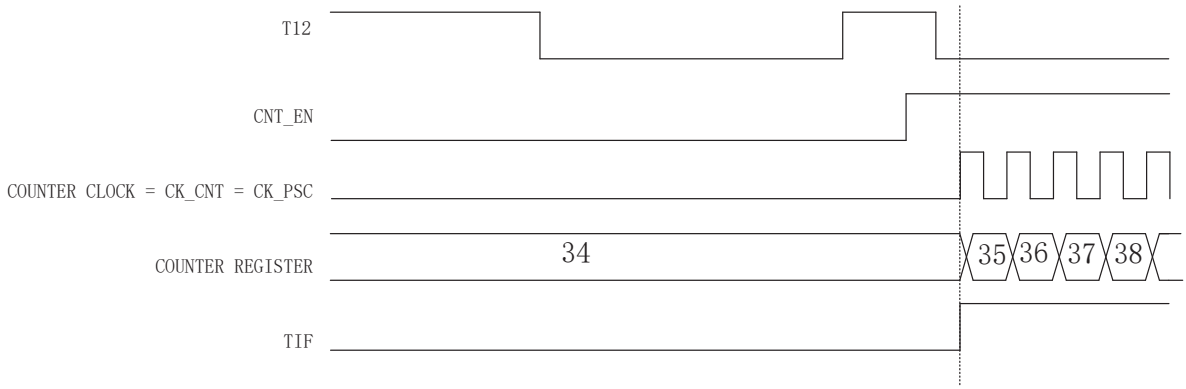


Figure 0.24 Control Circuit for Standard Trigger Mode

See Standard Trigger timing diagram below.

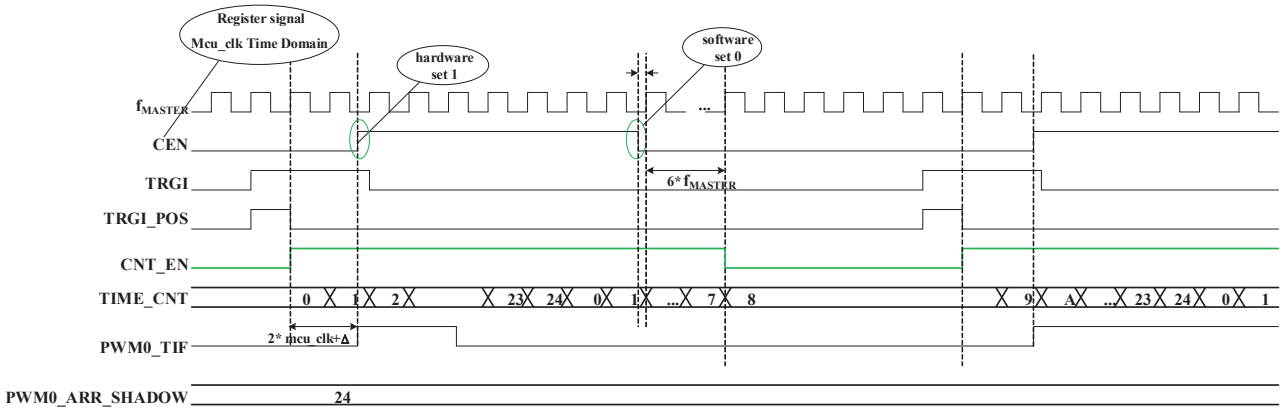


Figure 0.25 Standard Trigger Timing Diagram

8.1.3.5.2 Reset Trigger Mode

When a trigger input event occurs, the counter and its prescaler can be reinitialized; at the same time, an UEV is also generated if the URS bit of the PWM0_CR1 register is low; then all the preload registers (PWM0_ARR, PWM0_PSCR, PWM0_CCRx) are updated.

In the following examples, the rising edge on the TI1 input causes the up counter to be cleared to 0.

- Configure channel 1 to detect the rising edge of TI1. Configure the bandwidth of the input filter (in this example, no filter is needed, so keep IC1F=0000). The capture prescaler is not used in the trigger operation, so no configuration is required. The CC1S bit is only used to

select the input capture source which also does not need to be configured. Configure the PWM0_CCER1 register with CC1P=0 to select the polarity (rising edge detection only).

- Configure SMS=100 in PWM0_SMCR to select timer as reset trigger mode; configure TS=101 in PWM0_SMCR to select TI1 as input source.
- Configure CEN=1 of PWM0_CR1 register to start the counter.

The counter starts counting according to the internal clock until a rising edge appears on TI1; also, the counter is cleared and starts counting from 0 again. At the same time, the trigger flag (TIF bit in PWM0_SR1 register) is set and an interrupt request is generated if the interrupt is enabled (TIE bit in PWM0_IER register).

The delay between the rising edge of TI1 and the actual reset of the counter depends on the resynchronization circuit at the TI1 input. The figure below shows the behavior at the auto-reload register PWM0_ARR = 0x36.

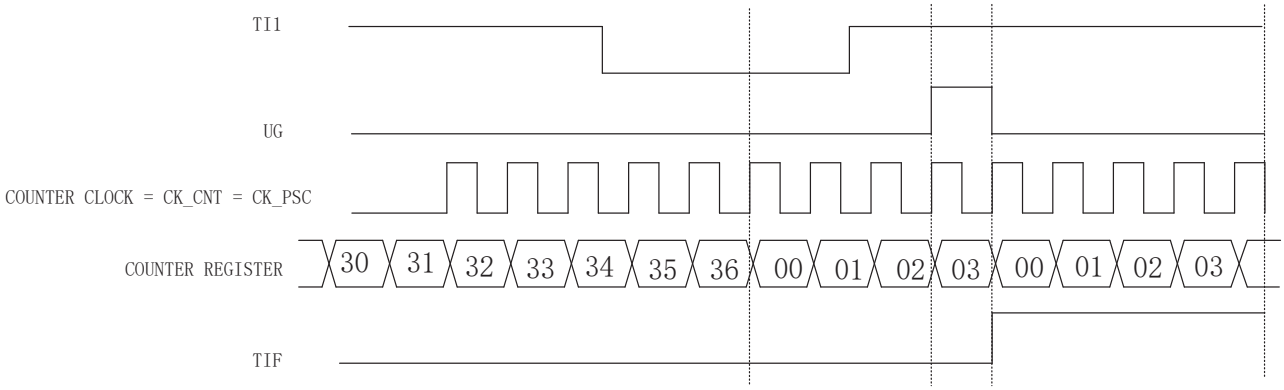


Figure 0.26 Control Circuit in Reset Trigger Mode

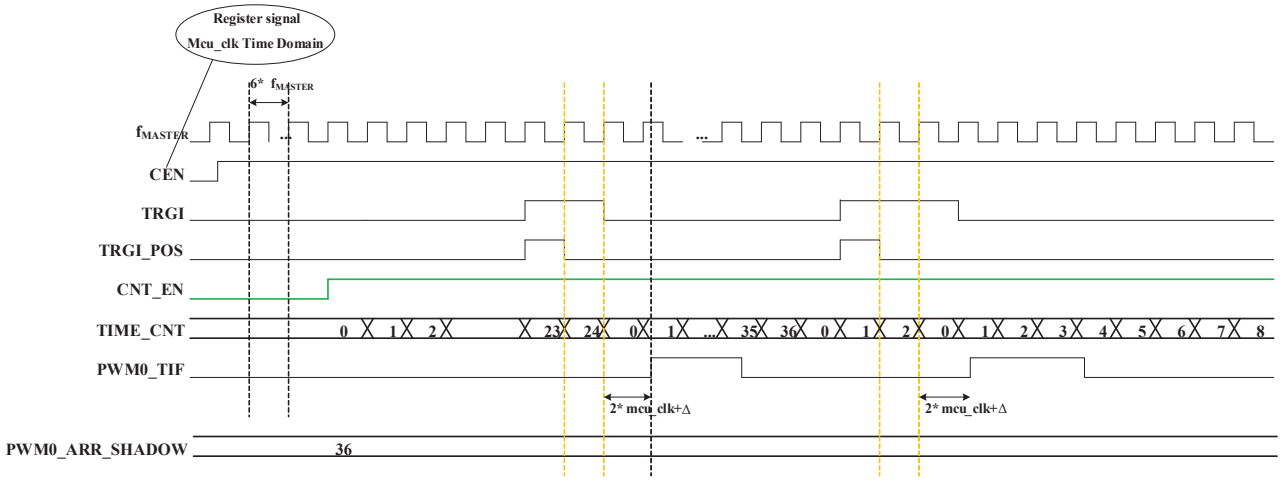


Figure 0.27 Reset Trigger Timing Diagram

8.1.3.5.3 Gate Trigger Mode

The counter is enabled by the current level of the selected input signal.

In the following example, the counter only counts up when TI1 is low:

1) Configure channel 1 for detecting the low level on TI1. Configure the input filter bandwidth (in this example, no filtering is required, so keep IC1F=0000). The capture prescaler is not needed in the trigger operation, so no configuration is required. The CC1S bit is used to select the input capture source and also does not need to be configured. Configure the PWM0_CCER1 register with CC1P=1 to determine the polarity (detect low level only).
2) Configure SMS=101 in PWM0_SMCR register to select timer as gate trigger mode; configure TS=101 in PWM0_SMCR register to select TI1 as input source.
3) Configure the PWM0_CR1 register with CEN=1 to start the counter (in gate control mode, if CEN=0, the counter cannot be started, regardless of the trigger input level).

The counter starts counting according to the internal clock as long as TI1 is low, and stops counting once TI1 becomes high. The TIF flag bit is set when the counter either starts or stops.

The delay between the rising edge of TI1 and the actual stop of the counter depends on the resynchronization circuit at the TI1 input.

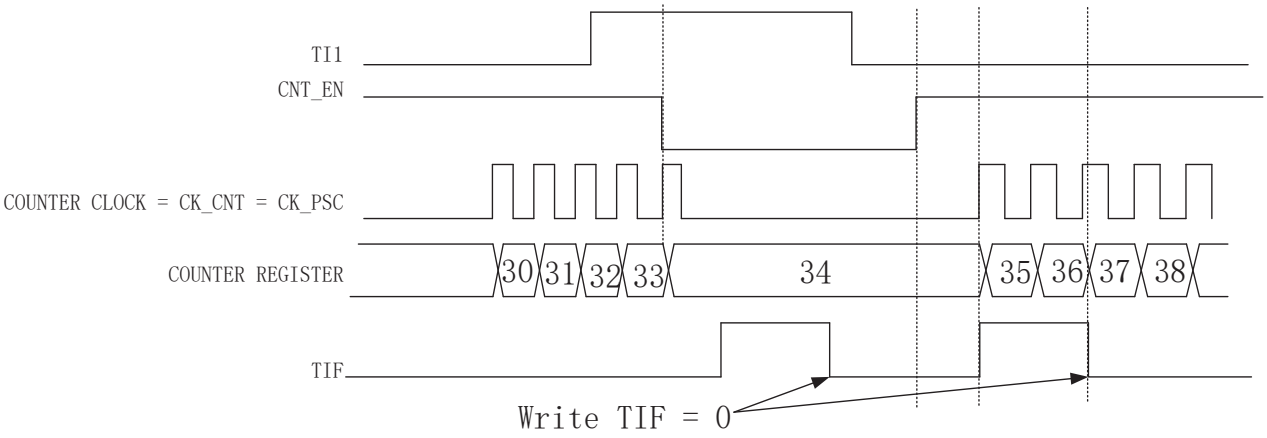


Figure 0.28 Control Circuit in Gate Trigger Mode

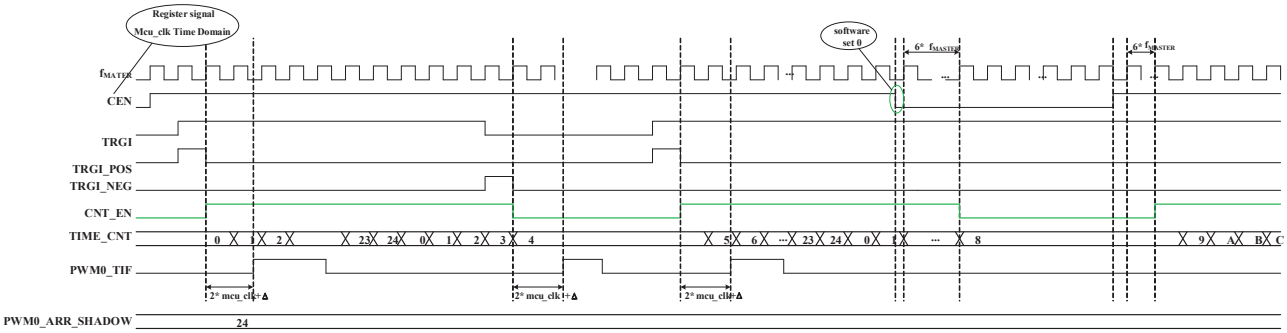


Figure 0.29 Gate Trigger Timing Diagram

8.1.3.5.4 External Clock Mode 2 + Trigger Mode

External clock mode 2 can be used with another input signal in trigger mode. In this case, the ETR signal is used as an input to the external clock and the other input signal can be used as a trigger mode (standard trigger mode, reset trigger mode and gate trigger mode are supported).

Note: The ETR signal cannot be configured as input signal TRGI at this time (via the TS bit in the PWM0_SMCR register).

In the following example, the counter counts up once on each rising edge of the ETR once a rising edge occurs on TI1.

- Configure the external trigger input circuit via the PWM0_ETR register. In this example, ETF = 0000 since no filtering is used. Configure ETPS = 00 to disable prescaler, ETP = 0 to monitor the rising edge of the ETR signal, and ECE = 1 to enable external clock mode 2.
- Use channel 1 to monitor the rising edge of TI1. Configure input filter (since no filter is required there, configure IC1F=0000). Since the trigger operation does not use prescaler, so there is no need to configure it and the CC1S bit is only used to select the input capture source and therefore does not need to be configured. Configure the PWM0_CCER1 register with CC1P=0 to select the rising edge trigger.
- Configure SMS=110 in PWM0_SMCR register to select timer as trigger mode. Configure TS=101 in PWM0_SMCR register to select TI1 as input source.

When a rising edge occurs on TI1, the TIF flag is set and the counter starts counting on the rising edge of ETR.

The delay between the rising edge of the TI1 signal and the actual counter clock depends on the resynchronization circuit at the TI1 input.

The delay between the rising edge of the ETR signal and the actual counter clock depends on the resynchronization circuit at the ETRP input.

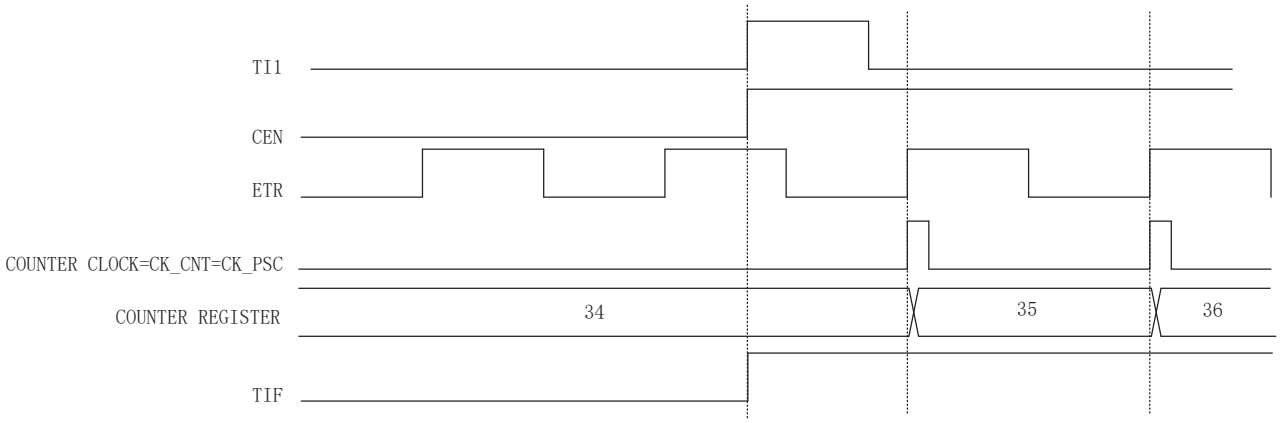


Figure 0.30 Control Circuit in External Clock Mode 2 + Trigger Mode

8.1.3.6 Sync with General Purpose Timer PWM1

Timers are interconnected inside the chip for synchronization or linking. When a timer is configured as master mode, it can output a trigger signal (TRGO) to those timers configured as slave mode to

complete reset, start, stop operations, or as a drive clock for those timers.

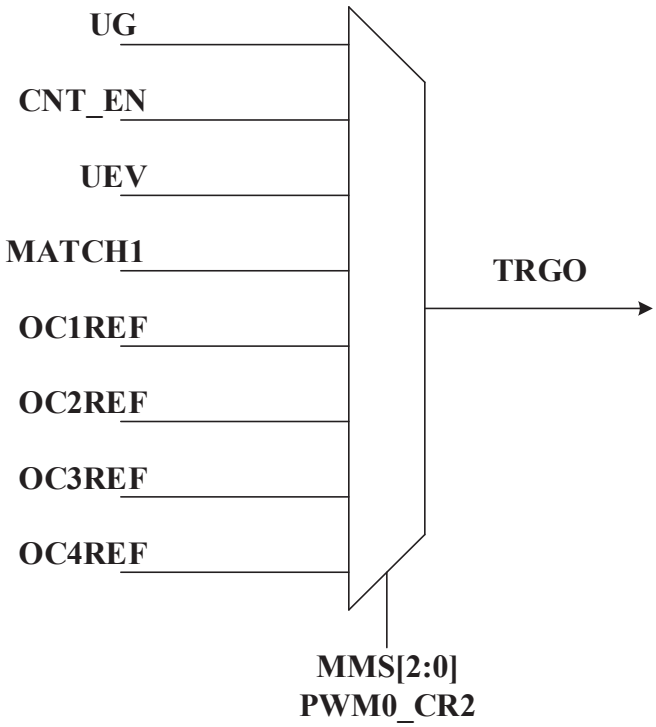


Figure 0.31 Trigger Master/Slave Mode Selection

8.1.3.6.1 Use a Timer as a Prescaler for Another Timer

Figure 8.32 shows an example of trigger Master/Slave Mode.

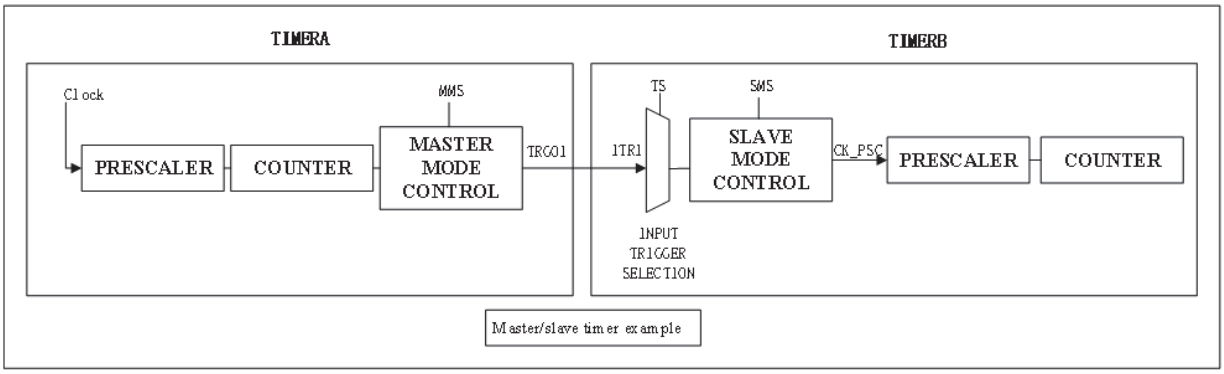


Figure 0.32 Master/Slave Connection

For example, the user can configure Timer A as a prescaler for Timer B, which requires configurations as follows:

1) Configure Timer A as the master mode so that a periodic trigger signal is output at each update event (UEV). Configure the PWMx_CR2 register with MMS=010 so that TRGO1 can output a rising edge at each update event.
2) The TRGO1 signal output from Timer A is linked to Timer B. Timer B needs to be configured to trigger slave mode, using ITR1 as the input trigger signal. The above-mentioned operation can be achieved by configuring the TS bit in the PWMx_SMCR register (configure the TS=100 in the PWM0_SMCR register).
3) Configuring SMS=111 in the PWMx_SMCR register sets the clock/trigger controller to external clock mode 1. This operation will enable the rising edge of the periodic trigger signal (generated by the overflow of Timer A) output from Timer A to drive the clock of Timer B.
4) Finally, set the CEN bit of both timers (in PWMx_CR1 register) to enable them.

Note: If OCx is selected as the trigger signal for Timer A output (MMS=1xx), its rising edge will drive the clock of Timer B.

8.1.3.6.2 Use One Timer to Enable Another Timer

Figure 8.33 shows that Timer B is enabled by the comparison output of Timer A. Please refer to the following diagram for the connection of the timers *Figure 0.32 Master/Slave Connection*. Timer B only counts according to its own drive clock when the OC1REF signal of Timer A is high. Both timers use a f_{MASTER} with prescaler factor of 4 as the clock ($f_{CK\_CNT} = f_{MASTER}/4$).

1) Configure Timer A as the master mode and output the comparison output signal (OC1REF) as the trigger signal. (Configure PWMx_CR2 register with MMS=100).
2) Configure the waveform of Timer A OC1REF signal (PWMx_CCMR1 register).
3) Configure Timer B to use the output of Timer A as its own trigger input signal (configure the PWMx_SMCR register with TS=100).
4) Configure Timer B as gate trigger mode (configure SMS=101 of PWMx_SMCR register).
5) Set the CEN bit (PWMx_CR1 register) to enable Timer B.
6) Set the CEN bit (PWMx_CR1 register) to enable Timer A.

Note:

The clocks of the two counters are not synchronized,which affects only the enable signal of Timer B.

Figure 8.33 shows Timer A output gate triggering Timer B.

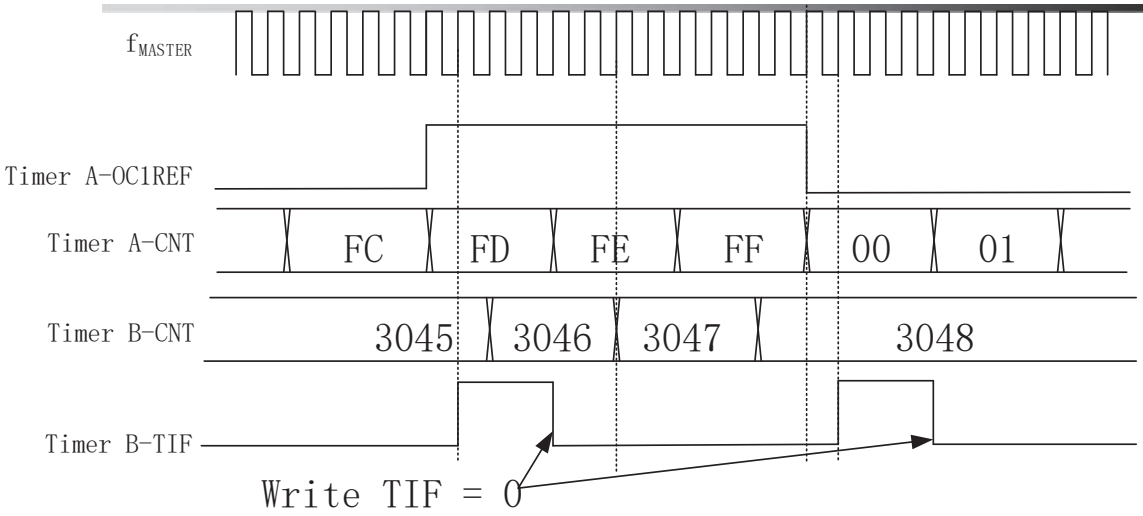


Figure 0.33 Timer A Output Gate Triggering Timer B

Both the counter and prescaler of Timer B above are not initialized before startup, so both start counting from the existing value. If the two timers are reset before starting Timer A, the user can write the desired value to the timer's counter to make it count from the specified value. The reset operation of the timers can be achieved by software writing the UG bit of the PWMx_EGR register.

1. In the next example, we synchronize Timer A and Timer B. Timer A is in master mode and starts counting from 0. Timer B is triggered slave mode and starts counting from 0xE7.

Both timers use the same prescaler factor. When the CEN bit of PWMx_CR1 register is cleared, Timer A is disabled and Timer B stops counting at the same time.

2. Configure Timer A as the master mode and the comparison output signal (OC1REF) as the trigger signal to output. (Configure PWMx_CR2 register with MMS=100).
3. Configure the waveform of OC1REF signal of Timer A (PWMx_CCMR1 register).
4. Configure Timer B to use the output of Timer A as its own trigger input signal (configure the PWMx_SMCR register with TS=100).
5. Configure Timer B as gate trigger mode (configure SMS=101 of PWMx_SMCR register).
6. Reset Timer A by writing 1 to the UG bit (PWMx_EGR register).
7. Reset Timer B by writing 1 to the UG bit (PWMx_EGR register).
8. Initialize Timer B by writing 0xE7 into Timer B counter (PWMx_CNTRL).
9. Enable Timer B by writing 1 to the CEN bit (PWMx_CR1 register).
10. Timer A is started by writing 1 to the CEN bit (PWMx_CR1 register).
11. Stop Timer A by writing 0 to the CEN bit (PWMx_CR1 register).

Figure 8.34 shows the counter enable signal (CNT_EN) of timer A gated to trigger timer B.

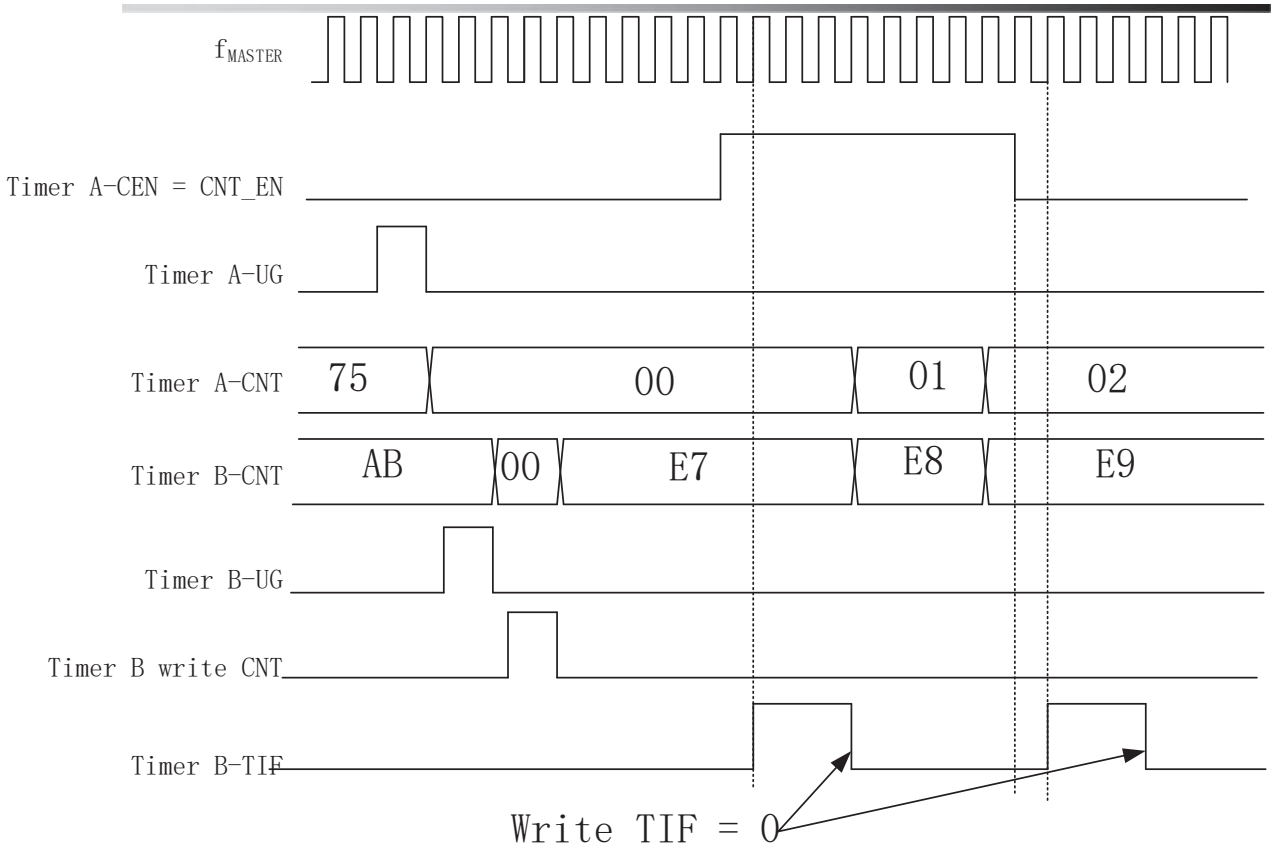


Figure 0.34 Counter enable signal (CNT_EN) of timer A gates to trigger timer B

8.1.3.6.3 Use One Timer to Start Another Timer

Timer B is enabled with an update event from Timer A. Timer B counts from its existing value (which can be a non-zero value) according to Timer B's own drive clock when an update event occurs from Timer A. Timer B automatically enables the CEN bit when it receives a trigger signal and continues until the user writes a 0 to the PWMx_CR1 register. Timer B automatically enables the CEN bit upon receipt of a trigger signal and starts counting until the user writes 0 to the CEN bit of the PWMx_CR1 register. Both timers use a f_{MASTER} with a prescaler factor of 4 as the drive clock ($f_{CK\_CNT} = f_{MASTER}/4$).

1) Configure Timer A as the master mode and output the update signal (UEV). (Configure PWMx_CR2 register with MMS=010).
2) Configure the period of timer A (PWMx_ARR register).
3) Configure Timer B to use the output of Timer A as the trigger signal for the input (configure the PWMx_SMCR register with TS=100).
4) Configure Timer B as trigger mode (configure SMS=110 of PWMx_SMCR register).
5) Set the CEN bit (PWMx_CR1 register) to start Timer A.

Figure 8.35 shows the timing diagram of Timer A update event (TIMERA-UEV) triggering Timer B.

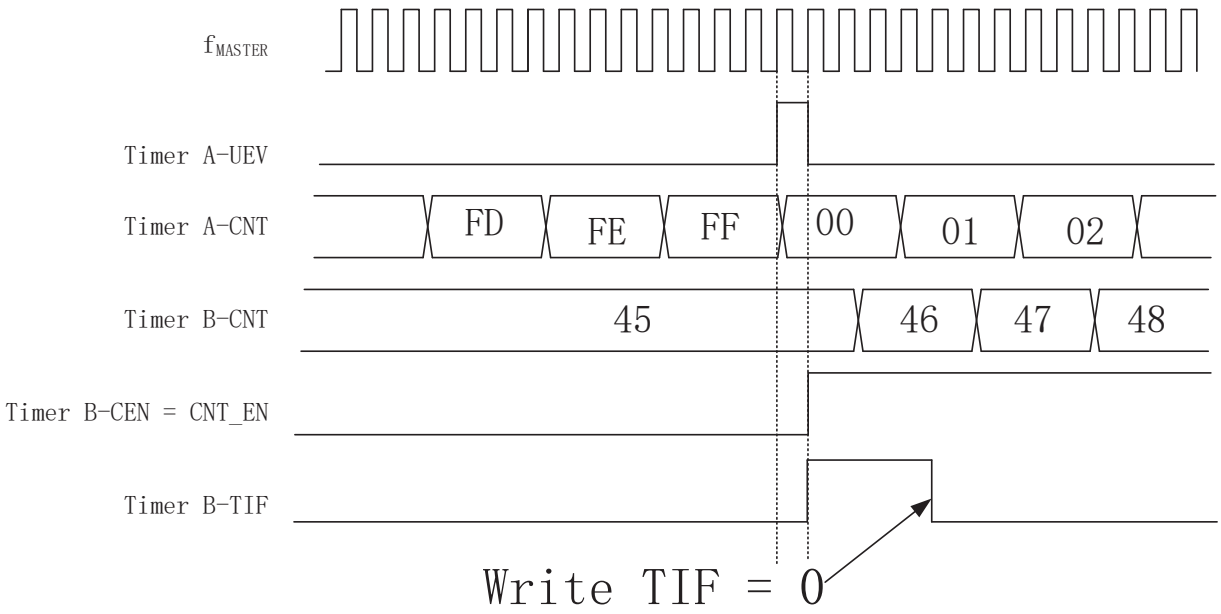


Figure 0.35 Timer A Update Event (TIMERA-UEV) Triggers Timer B

As in the previous example, the user can also initialize the counters before starting them. The example below uses the same configuration with *Figure 8.34 Counter enable signal (CNT_EN) of timer A gates to trigger timer B*, but with the normal trigger mode (configuring SMS=110 of PWMx_SMCR register) instead of the gated trigger mode.

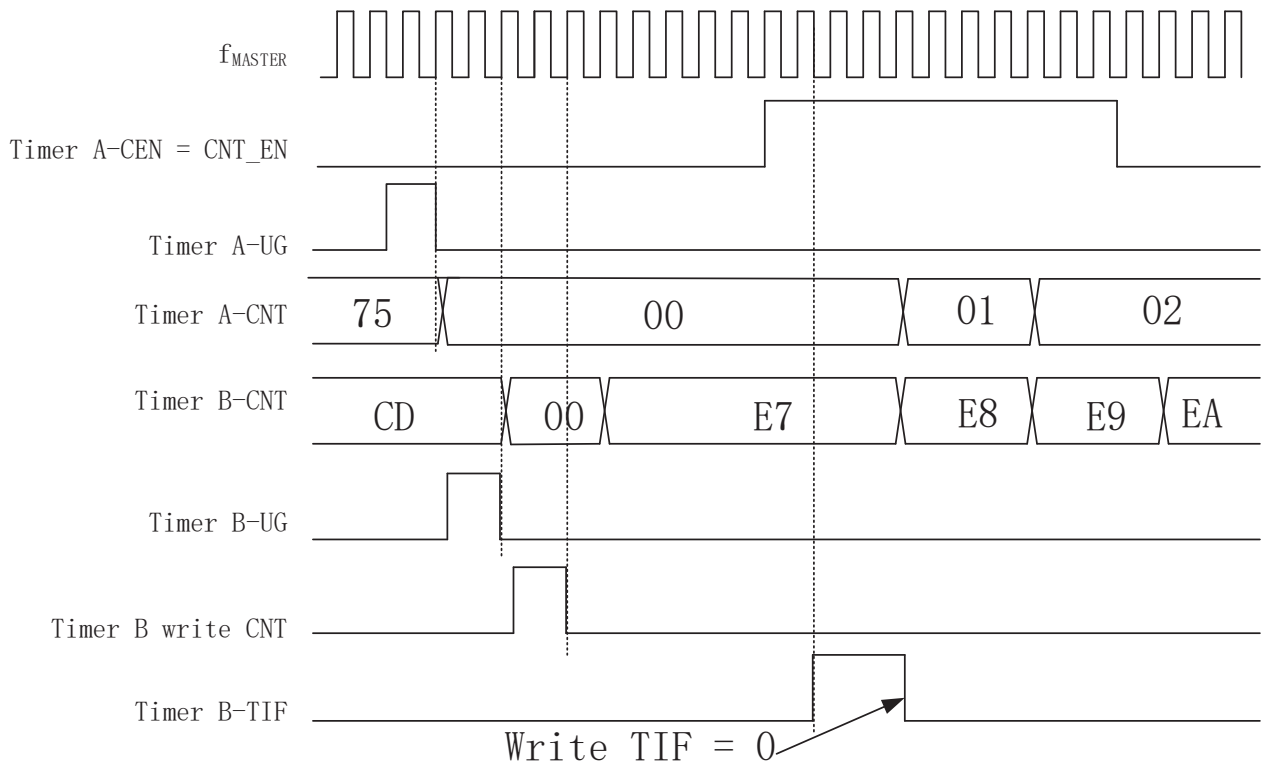


Figure 0.36 Counter enable signal CNT_EN of Timer A triggers Timer B

8.1.3.6.4 Trigger Two Timers Synchronously with External Signals

Timer A is enabled using the rising edge of TI1, and Timer B is enabled at the same time. Timer is connected as shown in *Figure 0.32* Master/Slave Connection. To maintain the timer link, Timer A needs to be configured as master/slave mode (slave mode for TI1 signal and master mode for Timer B).

1) Configure Timer A as the master mode and output enable signal as the trigger of Timer B (configure MMS=001 of PWMx_CR2 register).
2) Configure Timer A as slave mode and use TI1 signal as the input trigger signal (configure the PWMx_SMCR register with TS=100).
3) Configure the trigger mode of Timer A (configure SMS=110 of PWMx_SMCR register).
4) Configure Timer A as master/slave mode (configure MSM=1 of PWMx_SMCR register).
5) Configure Timer B to use the output of Timer A as the input trigger signal (configure the PWMx_SMCR register with TS=100).
6) Configure the trigger mode of Timer B (configure SMS=110 of PWM0_SMCR register).

When a rising edge occurs on TI1 (Timer A's), both timers start counting synchronously and the TIF bits are set.

Note: In this example, both timers are initialized (set UG bit) before starting, so they both start counting from 0. However, the user can also insert an offset by modifying the counter register (PWMx_CNT), in which case a delay will be inserted between the CK_PSC signal and the CNT_EN signal in timer A.

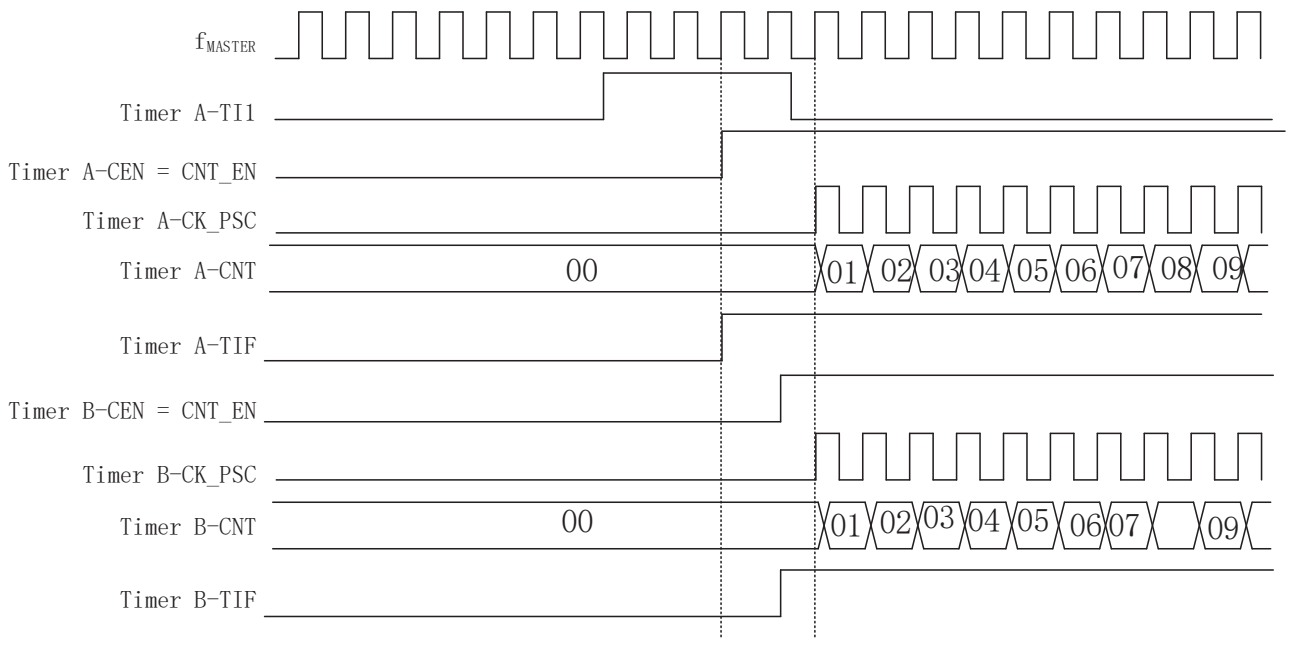


Figure 0.37 TI1 Signal from Timer A triggers Timer A and Timer B

8.1.4 Capture/Compare Channels

The timer I/Os (PWM0_CCx) can be used as input capture or output compare. This function can be implemented by configuring the CCxS channel selection bit of Register (PWM0_CCMRx) in Capture/Compare Channel Mode, where the x represents the number of channels.

Each capture/compare channel is built around a capture/compare register (containing shadow registers), including the capture input (digital filter, multiplexer, and prescaler), and capture output (comparator and output control).

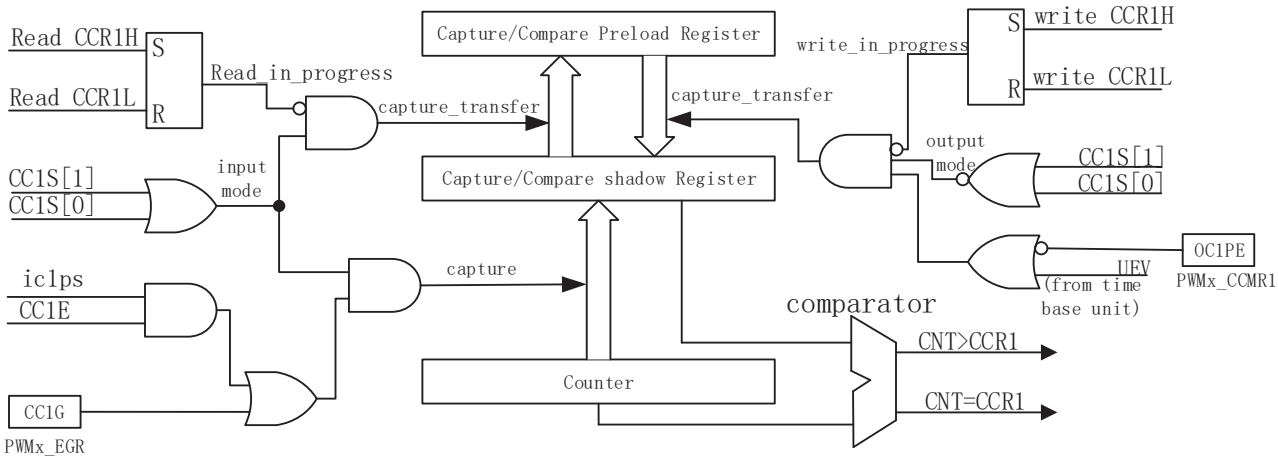


Figure 0.38 Main Circuit for Capture/Compare Channel 1

The capture/compare module consists of a preload register and a shadow register. The read/write operates only on the preload register.

In capture mode, the capture occurs on the shadow register and is then copied to the preload register.

In compare mode, the contents of the preload register are copied to the shadow register, and then the contents of the shadow register are compared with the counter.

When the channel is configured as output mode (CCxS=0 in PWM0_CCMRx register), the PWM0_CCRx register can be accessed at any time. (The x here refers to the number of channels)

When the channel is configured as input mode, the read operation on the PWM0_CCRx register is similar to the counter read operation (See Figure 8.39). When a capture occurs, the counter contents are captured to the PWM0_CCRx shadow register and later copied to the preload register. The preload register is frozen while the read operation is in progress.

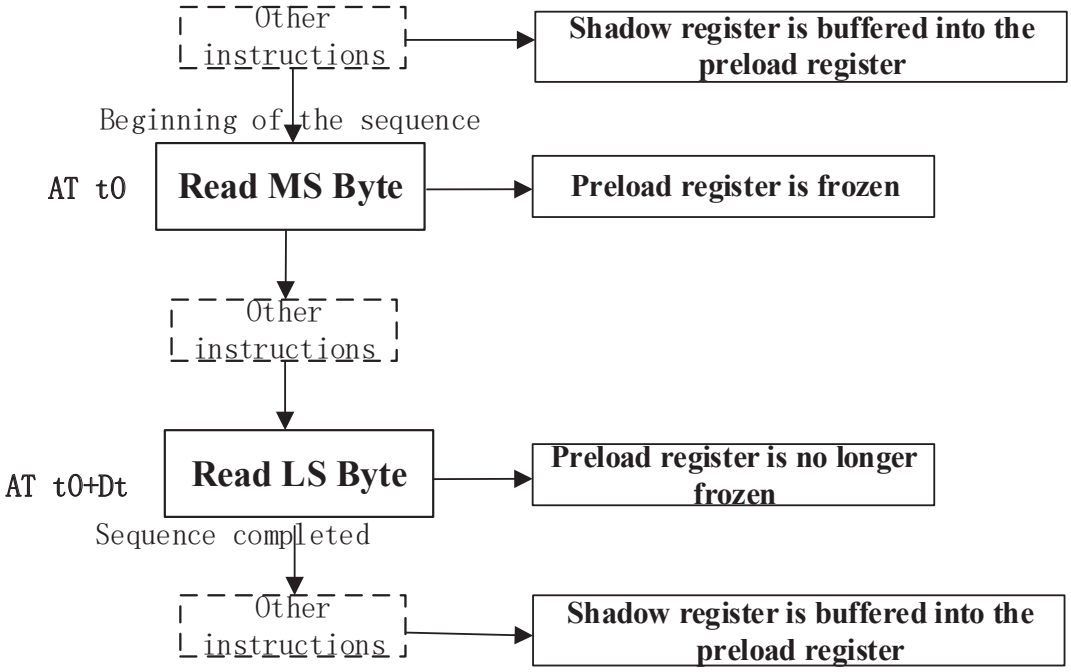


Figure 0.39 Read Operation of the 16-bit PWM0_CCRx register in capture mode

Figure 8.39 demonstrates how the 16-bit CCRx register reads. The cached data will remain unchanged until the read operation ends.

At the end of the whole read process, if only the PWM0_CCRxL register is read, the low bit (LS) of the counter value is returned.

If the high (MS) data is read after the low (LS) data, this low data will not be returned.

8.1.4.1 Read/Write the 16-bit PWM0_CCRx Register

When the channel is configured as output mode, the write operation of the 16-bit PWM0_CCRx register is implemented through the preload register. The high byte must be written first. While writing the high byte, shadow register update is disabled until the low byte write operation is completed, which is similar to PWM0_ARR register write operation.

The read operation of the PWM0_CCRx register is implemented through the pre-load register.

When the channel is configured as input mode, the PWM0_CCRx register is read-only at this time. The read operation of the PWM0_CCRx register is similar to that of the counter. When a capture occurs, the contents of the counter are captured into the PWM0_CCRx shadow register and later copied into the preload register. The preload register is frozen while the read operation is in progress.

8.1.4.2 Input Module

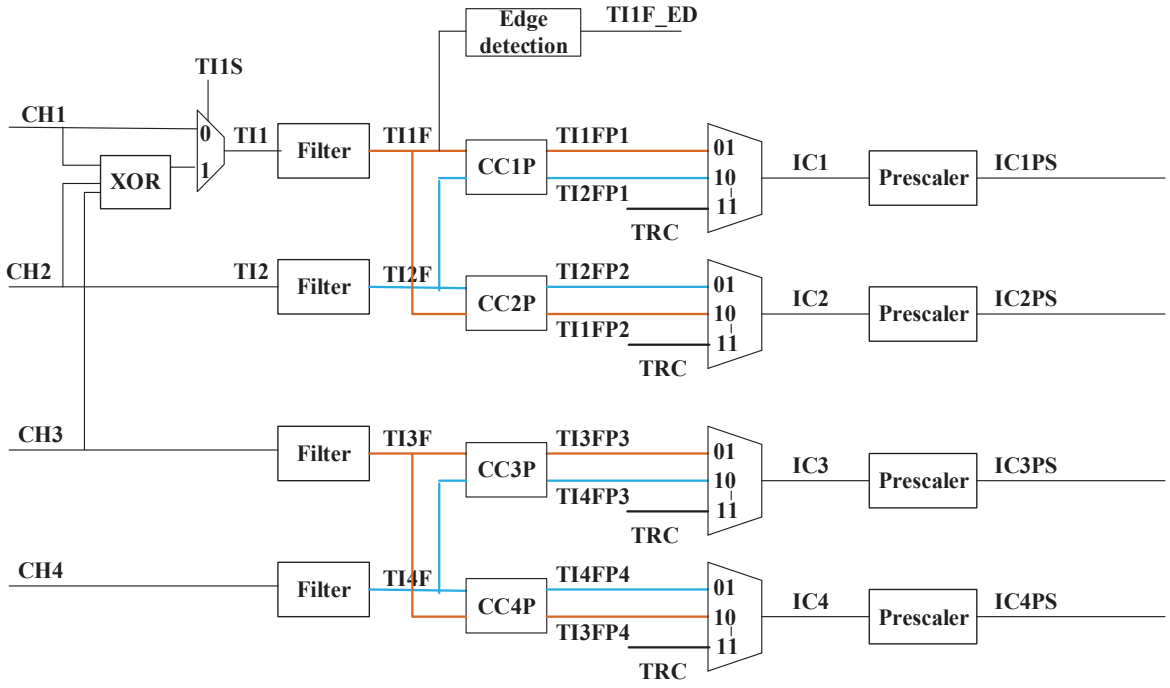


Figure 0.40 Input Module Block Diagram

As shown below, input the corresponding TIx input signal to sample and generates a filtered signal TIxF. An edge monitor with polarity selection then generates a signal (TIxFPx) that can be used as an input trigger for the trigger mode controller or as a capture control. This signal is prescaled into a capture register (ICxPS).

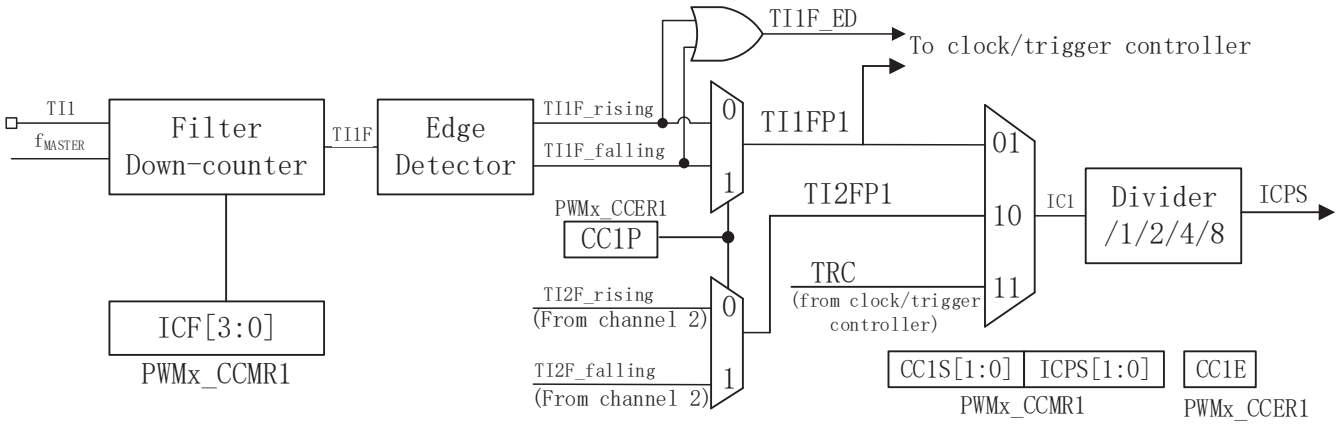


Figure 0.41 Inputs of PWM0 Channel 1

8.1.4.3 Input Capture Mode

In the input capture mode, the present value of the counter is latched into the capture/compare register (PWM0_CCRx) when the corresponding edge on the ICx signal is detected. When a capture event occurs, the corresponding CCxIF flag (PWM0_SR register) is set to 1.

If the CCxIE bit of the PWM0_IER register is set, which means an interrupt is enabled, then an interrupt request will be generated. If the CCxIF flag is already high when a capture event occurs, the repeat capture flag CCxOF (PWM0_SR2 register) is set to 1. Writing CCxIF=0 or reading the capture data stored in the PWM0_CCRxL register clears CCxIF. Writing CCxOF=0 clears CCxOF.

The examples below describe how to capture the counter value to the PWM0_CCR1 register on the rising edge of the TI1 input. See steps as follows:

1) Select a valid input: for example, PWM0_CCR1 is connected to the TI1 input, so write CC1S=01 to the PWM0_CCMR1 register, at which point the channel is configured as an input and the PWM0_CCR1 register becomes read-only.
2) According to the characteristics of the input signal TIx, the filtering time of the corresponding input filter can be set by configuring the ICxF bit in the PWM0_CCMRx register. Assuming that the input signal is jittered for a maximum of 5 clock cycles, the bandwidth of the filter shall be configured to be longer than 5 clock cycles; therefore, it can be sampled 8 times in succession to confirm a true edge transition on TI1, i.e., write IC1F=0011 in the PWM0_CCMR1 register, at which point, the signal is valid only if 8 identical TI1 signals are sampled in succession (sampling frequency is f_{MASTER}).
3) Select the valid conversion edge of TI1 channel and write CC1P=0 (rising edge) in PWM0_CCER1 register.
4) Configure the input prescaler. In this example, we want the capture to occur at every active level transition moment, so the prescaler is disabled (write IC1PS=00 in PWM0_CCMR1 register).
5) Set CC1E=1 in PWM0_CCER1 register to allow the counter value captured into the capture register.
6) If required, allow the associated interrupt request by setting the CC1IE bit in the PWM0_IER register.

When an input capture occurs,

- When a active level transition is generated, the counter value is transferred to the PWM0_CCR1 register.
- The CCxIF flag is set (interrupt flag). CCxOF is also set to 1 when at least 2 consecutive captures occur and CCxIF has not been cleared.
- If the CCxIE bit is set, an interrupt will be generated.

Note:

- To handle overcapture (CC1OF bit), it is recommended to read the data before reading the repetition capture flag, which is to avoid losing repetition capture information that may

be generated after reading the overcapture flag and before reading the data.

- Setting the corresponding CCxG bit in the PWM0_EGR register allows the input capture interrupt to be generated by software.
- The PWM0_CCMRx register is configured in two steps: the first configuration of the CCxS bit in the PWM0_CCMRx register to select the input mode mapping; the second configuration of the ICxF and ICxPSC in the PWM0_CCMRx register to select the filter and prescaler coefficients.

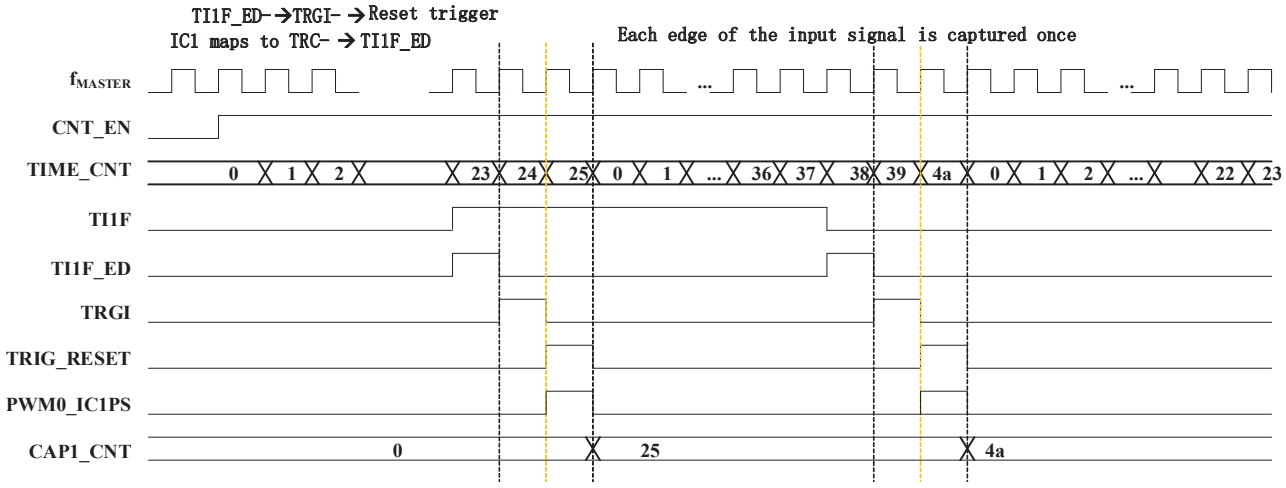


Figure 0.42 Input Capture Sequence

8.1.4.3.1 Measure the PWM Input Signal

This mode is a special case of the input capture mode and the operation is the same as the input capture mode except for the following differences:

1) Both ICx signals are mapped to the common TIx input.
2) The polarity of the active edges of these two ICx signals is reversed.
3) One of the TIxFP signals is used as the trigger input signal, and the trigger mode controller is configured to reset the trigger mode.

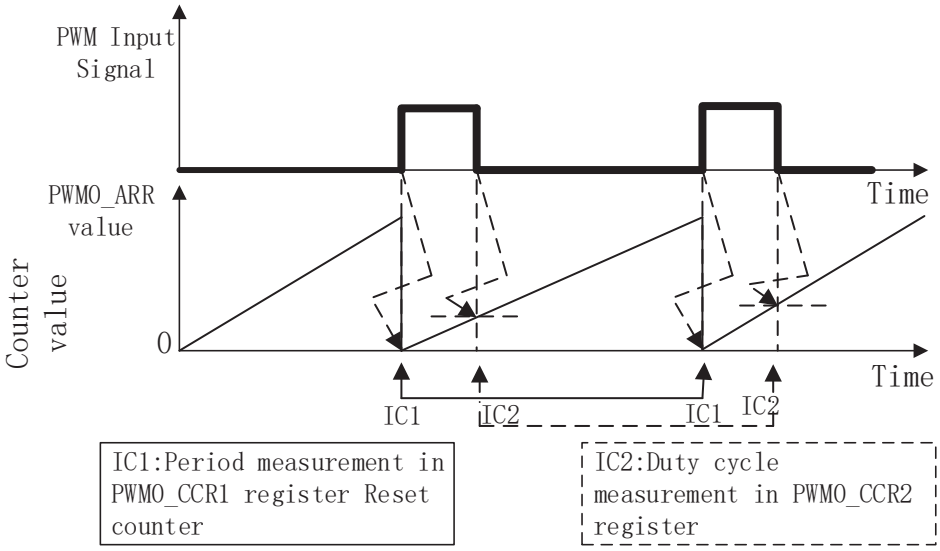


Figure 0.43 Measure PWM Input Signal

For example, you can measure the period (PWM0_CCR1 register) and duty cycle (PWM0_CCR2 register) of the PWM signal input on TI1 in the following way. (depending on the frequency of the f_{MASTER} and the value of prescaler)

1) Select active input for PWM0_CCR1: Set CC1S=01 in PWM0_CCMR1 register (TI1FP1 is selected).
2) Select the active polarity for TI1FP1 (used to capture data into PWM0_CCR1 and clear the counter): Set CC1P=0 (active on rising edge).
3) Select active input for PWM0_CCR2: Set CC2S=10 in PWM0_CCMR2 register (TI1FP2 is selected).
4) Select the active polarity of TI1FP2 (capture data to PWM0_CCR2): Set CC2P=1 (active on falling edge).
5) Select a valid trigger input signal: Set TS=101 in PWM0_SMCR register (TI1FP1 is selected).
6) Configure the trigger mode controller in reset trigger mode: Set SMS=100 in PWM0_SMCR.
7) Enable capture: Set CC1E=1 and CC2E=1 in PWM0_CCER1 register.

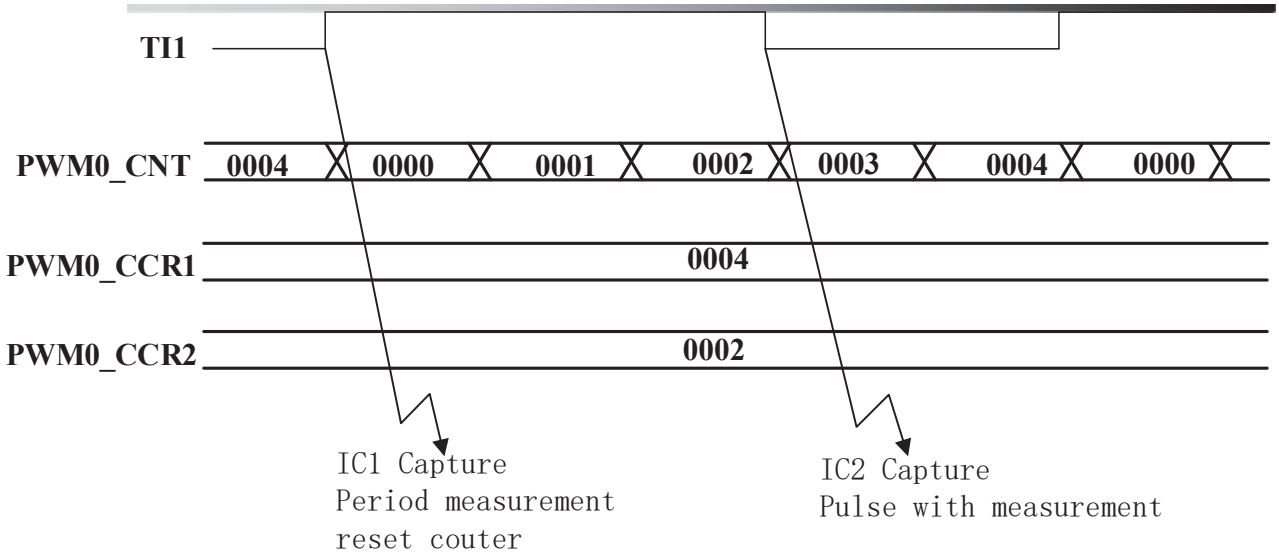


Figure 0.44 Example of PWM input signal measurement

Note:

1. To handle overcapture event (CCxOF bit), it is recommended to read the data before reading the repetition capture flag, this is to avoid losing repetition capture information that may be generated after reading the overcapture flag and before reading the data.
2. Input capture interrupts can be generated by software by setting the corresponding CCxG bits in the PWM0_EGR register.
3. Configure the PWM0_CCMRx register in two steps: first, configure the CCxS bit in the PWM0_CCMRx register to select the input mode mapping; second, configure the ICxF and ICxPSC in the PWM0_CCMRx register to select the filter and prescaler coefficients.

8.1.4.4 Output Module

The output module generates an intermediate waveform used for reference, called OCxREF (active high). Break functions and polarity act at the end of the chain.

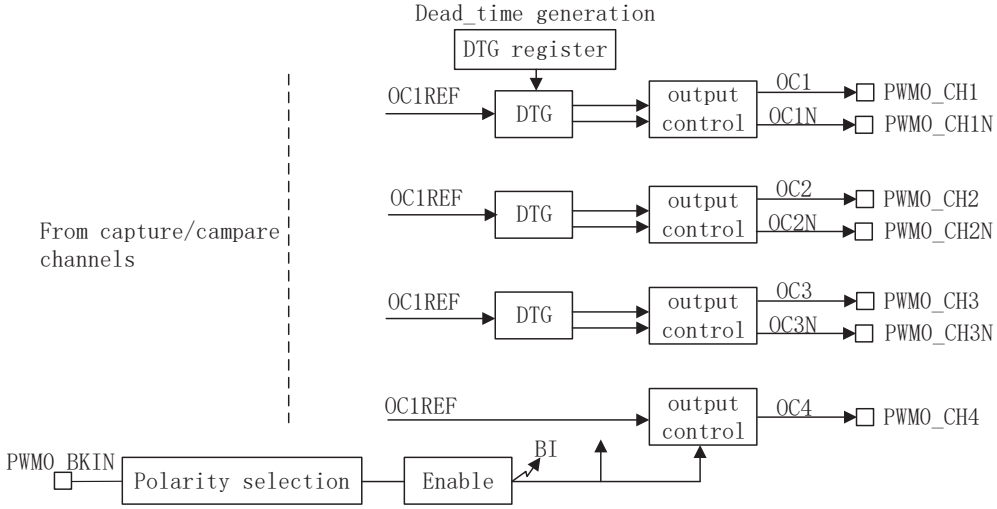


Figure 0.45 Output Module Diagram

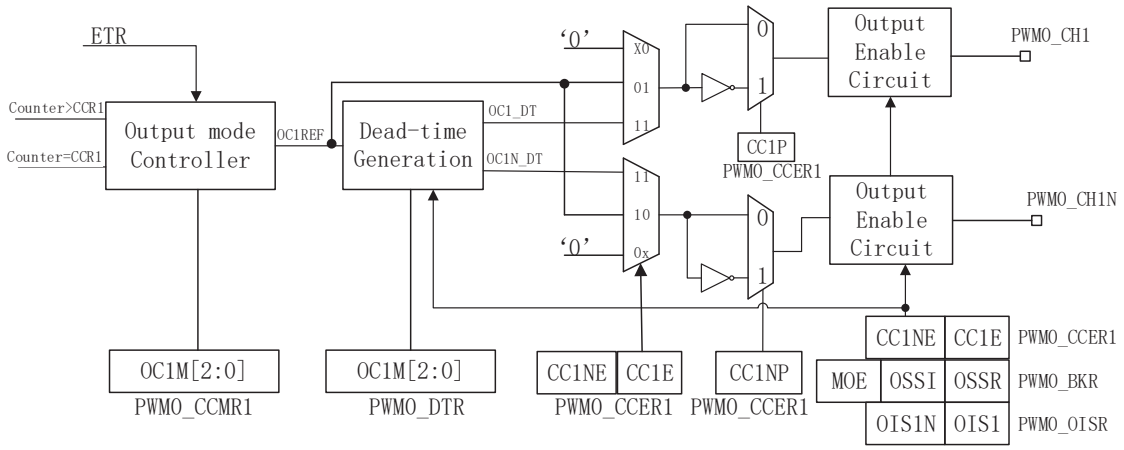


Figure 0.46 Detailed block diagram of the output module with complementary outputs (channel 1)

8.1.4.5 Forced Output Mode

In output mode (CCxS=0 in the PWM0_CCMRx register), the output compare signal can be forced by software to a high or low state, independent of the result of the comparison between the output compare register and the counter.

The output comparison signal is forced to be active by setting the corresponding OCxM=101 in the PWM0_CCMRx register. Thus, OCxREF is forced as high level (OCxREF is always active high), and the output of OCx is high or low depending on the CCxP polarity flag bit.

For example, if CCxP=0 (OCx is active high level), then OCx is forced to high level.

Set OCxM=100 in PWM0_CCMRx register to force the OCxREF signal to low.

In this mode, the comparison between the PWM0_CCRx shadow register and the counter is still ongoing, the corresponding flags are modified and the corresponding interrupts are still generated as well.

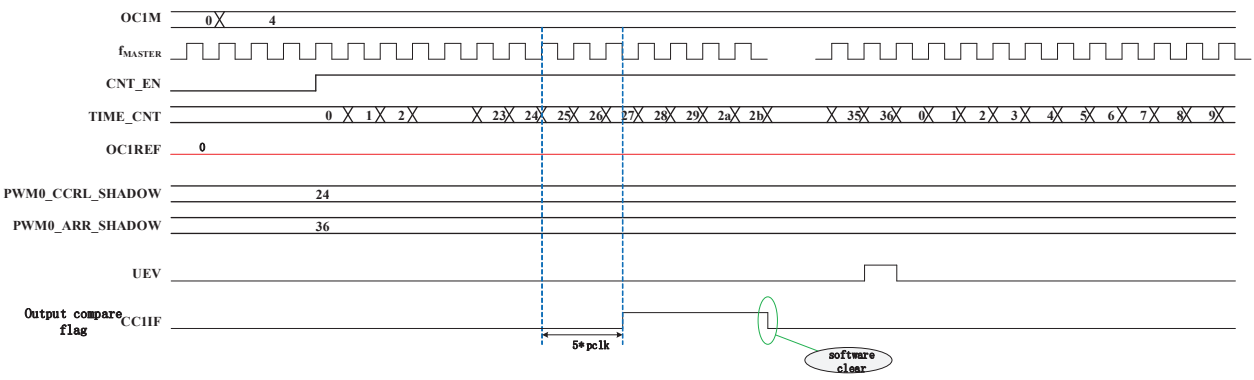


Figure 0.47 Forced Output inactive level

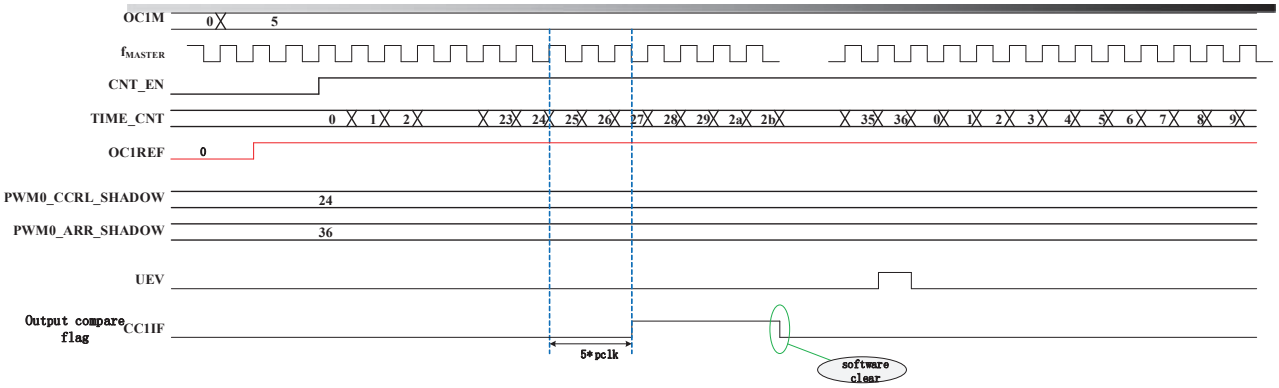


Figure 0.48 Forced Output Active Level

8.1.4.6 Output Compare Mode

This mode is used to control an output waveform or to indicate that a given period of time has been elapsed.

When a match is found between the capture/compare register and the counter (CCRx), execute the following operations:

- According to the different output compare modes, the corresponding OCx output signals:
  - Hold (OCxM=000)
  - Set to active level (OCxM=001)
  - Set to inactive level (OCxM=010)
  - Flip (OCxM=011)
- Set the flag bit in the interrupt status register (CCxIF bit in the PWM0_SR1 register).
- If the corresponding interrupt enable bit (CCxIE bit in PWM0_IER register) is set, an interrupt is generated.

The OCxM bit of the PWM0_CCMRx register is used to select the output compare mode, while the CCxP bit of the PWM0_CCMRx register is used to select the active and inactive level polarity.

The OCxPE bit of the PWM0_CCMRx register is used to select whether the PWM0_CCRx register needs to use the preload register or not.

In the output compare mode, the update event has no effect on the OCxREF and OCx outputs. The timing resolution is one cycle of the counter. The output compare mode can also be used to output a single pulse.

Configuration steps for output compare mode.

1. Selects the counter clock (internal, external, prescaler).
2. Write the corresponding data into the PWM0_ARR and PWM0_CCRx registers.
3. If an interrupt request is to be generated, set the CCxIE bit.
4. Select output mode steps:
   - Toggle the output pin of OCxM when the counter is required to match CCRx, set OCxM=011

— Disable preload register by writing OCxPE=0
— Write CCxP=0 to select high level as active level
— Set CCxE=1 to enable output

5. Set the CEN bit of the PWM0_CR1 register to start the counter

The PWM0_CCRx register can be updated by software at any time to control the output waveform, provided that the pre-load register is not used (OCxPE='0', otherwise the shadow register of PWM0_CCRx can only be updated when the next update event occurs). The following figure shows an example.

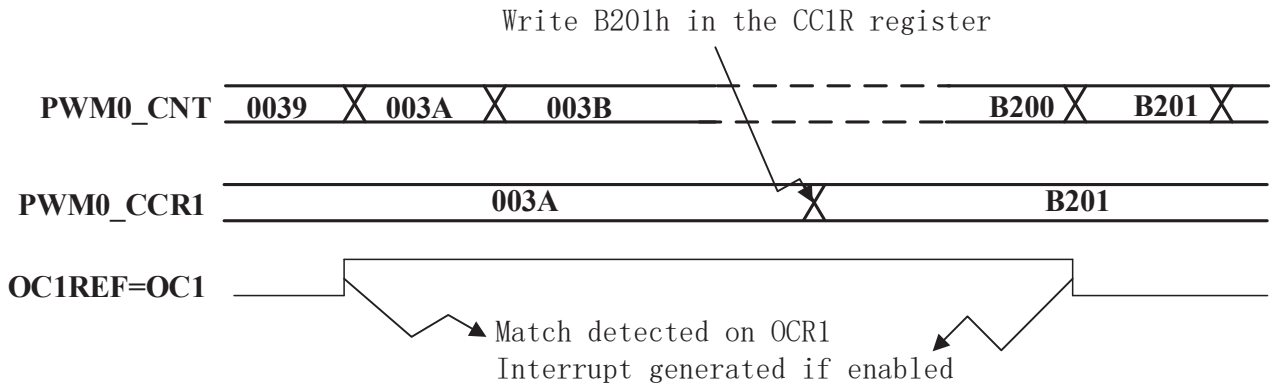


Figure 0.49 Output Compare Mode: flip-flop of OC1

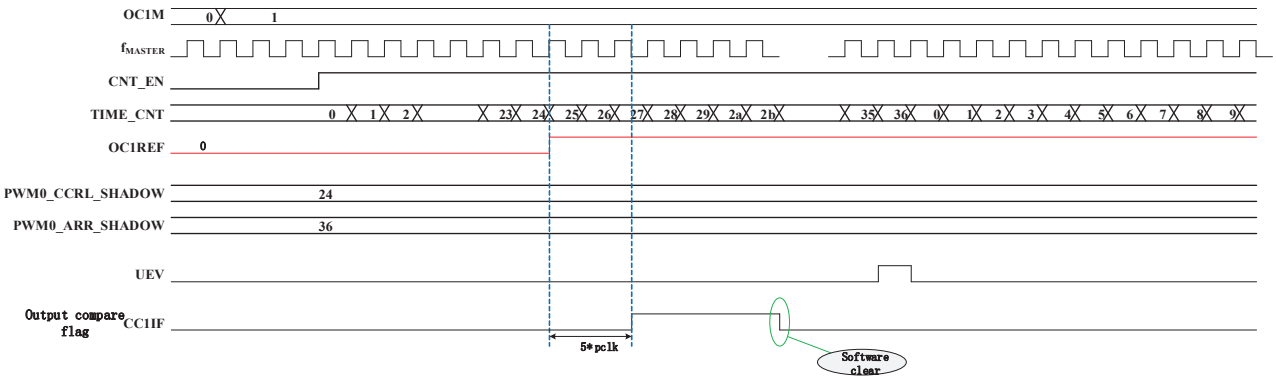


Figure 0.50 Output Compare (OCxM=0001)

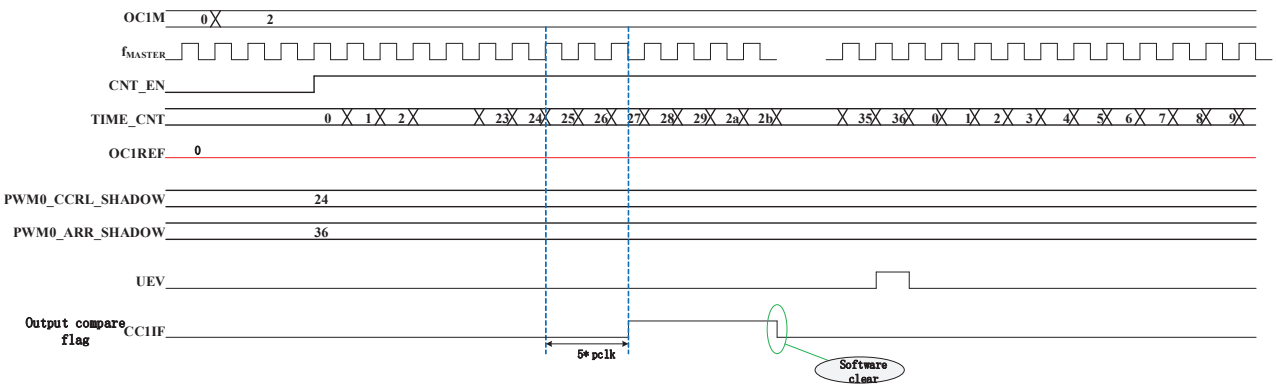


Figure 0.51 Output Compare (OCxM=0010)

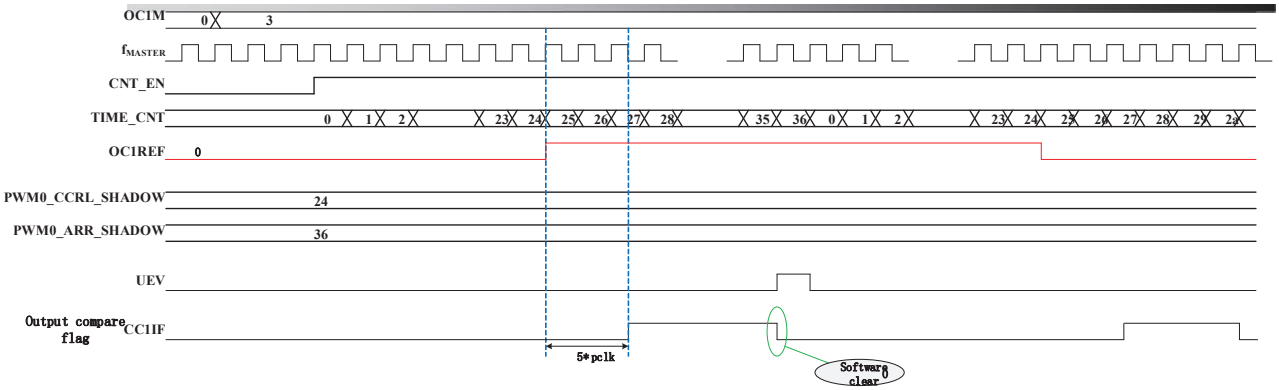


Figure 0.52 Output Compare (OCxM=0011)

8.1.4.7 PWM Mode

The pulse width modulation (PWM) mode generates a signal with a frequency determined by the PWM0_ARR register and a duty cycle determined by the PWM0_CCRx register.

The PWM mode can be selected independently on each channel (one PWM per OCx output) by writing 110 (PWM mode 1) or 111 (PWM mode 2) in the OCxM bits in the PWM0_CCMRx registers. The OCxPE bit in the PWM0_CCMRx register must be set to enable the corresponding preload register, or the ARPE bit of the PWM0_CR1 register can be set to enable the preload register for automatic reload (in up-counting mode or center aligned mode).

Note: Since the preload registers can be passed to the shadow registers only when an update event occurs, all registers must be initialized by setting the UG bit of the PWM0_EGR register before the counter starts counting.

The polarity of OCx can be set by software using the CCxP bit in the PWM0_CCERx register, which can be set to active high or active low. The output enable of OCx is controlled by a combination of CCxE, MOE, OISx and OSSR bits and OSSI bits in the PWM0_CCERx and PWM0_BKR registers. See the description of the PWM0_CCERx register for details.

In PWM mode (mode 1 or mode 2), PWM0_CNT and PWM0_CCRx are always being compared(based on the counting direction of the counter) to determine whether PWM0_CCRx $\leqslant$ PWM0_CNT or PWM0_CNT $\leqslant$ PWM0_CCRx is met.

According to the state of the CMS bit field in the PWM0_CR1 register, the timer is capable of generating either edge-aligned PWM signals or center-aligned PWM signals.

8.1.4.7.1 PWM Edge Aligned Mode

- Up-counting configuration

Up-counting when the DIR bit in the PWM0_CR1 register is low.

The following is an example of PWM mode 1. In PWM mode 1, when PWM0_CNT< PWM0_CCRx, the PWM reference signal OCxREF is high, otherwise it is low. If the comparison value in PWM0_CCRx is greater than the Auto-reload value (PWM0_ARR),

OCxREF remains '1'. If the compare value is 0, OCxREF remains '0'.

In PWM mode 2, the PWM reference signal OCxREF is low when PWM0_CNT< PWM0_CCRx, otherwise it is high. If the compare value in PWM0_CCRx is greater than the Auto-reload value (PWM0_ARR), OCxREF remains '0'. If the compare value is 0, OCxREF remains '1'.

Figure 8.53 shows an example of the edge-aligned PWM waveform when PWM0_ARR=8.

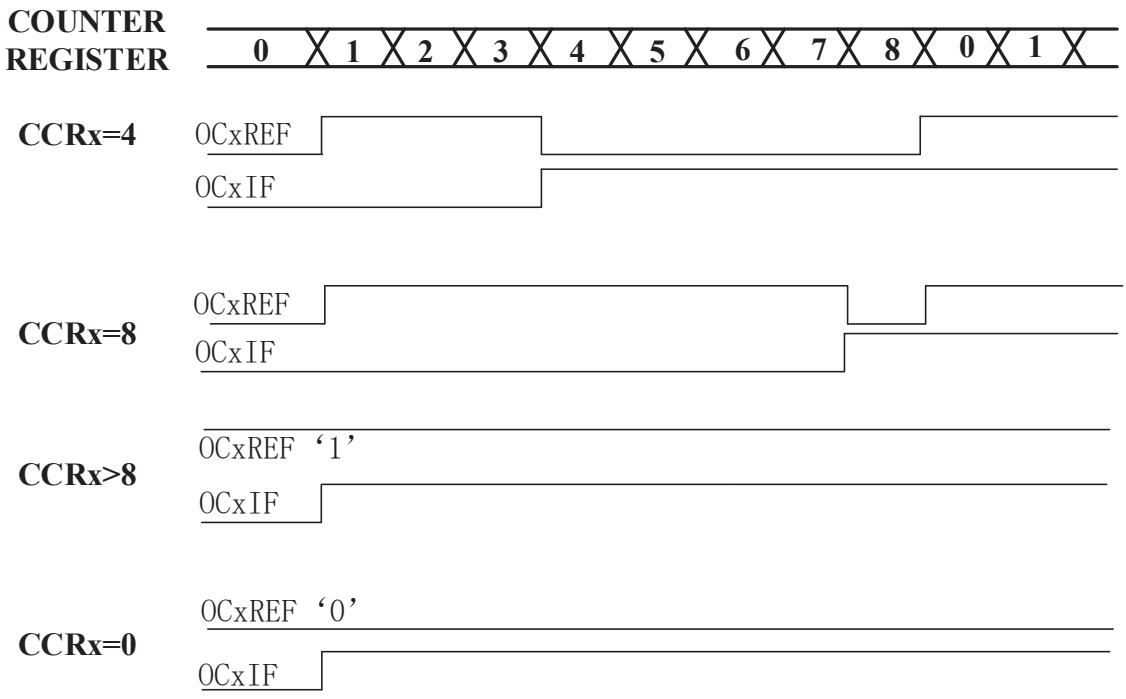


Figure 0.53 Edge-aligned, PWM mode 1 waveform (ARR=8)

- Down-counting configuration

  Down-counting is active when the DIR bit in the PWM0_CR1 register is high. Please refer to *Down-counting Mode*.

In PWM mode 1, the reference signal OCxREF is low when PWM0_CNT> PWM0_CCRx, otherwise it is high. If the comparison value in PWM0_CCRx is greater than or equal to the auto-reload value in PWM0_ARR, OCxREF remains '1'. A 0% PWM waveform cannot be generated in this mode.

In PWM mode 2, the reference signal OCxREF is high when PWM0_CNT> PWM0_CCRx, otherwise it is low. If the comparison value in PWM0_CCRx is greater than or equal to the auto-reload value in PWM0_ARR, OCxREF remains '0'. This mode cannot generate 100% PWM waveform.

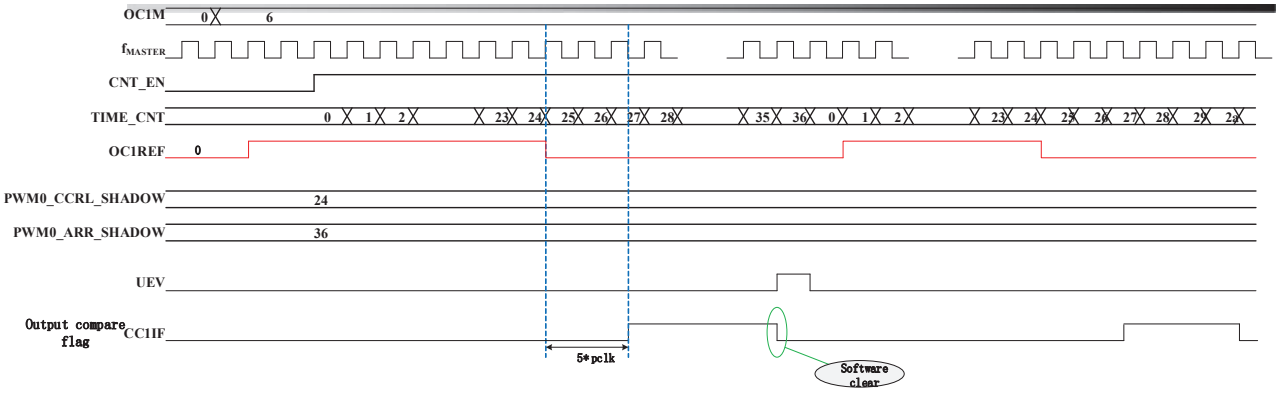


Figure 0.54 PWM Edge Aligned Mode 1 (Up-counting Configuration)

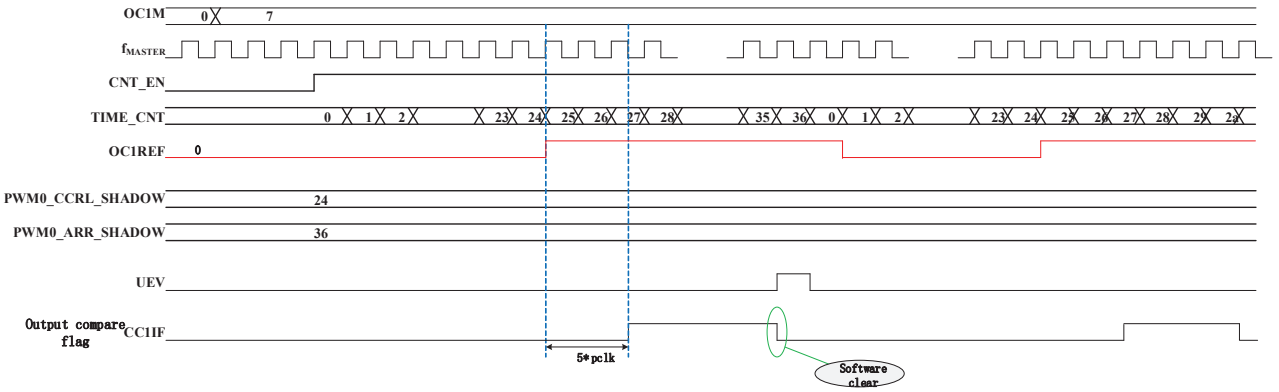


Figure 0.55 PWM Edge Aligned Mode 2 (Up-counting Configuration)

8.1.4.7.2 PWM Central Aligned Mode

Central aligned mode is enabled when the CMS bit in the PWM0_CR1 register is not '00' (all other configurations have the same effect on the OCxREF/OCx signals).

Depending on the setting of CMS bits, the compare flag can be set to 1 when the counter counts up, counts down, or counts up and down. The count direction bit (DIR) in the PWM0_CR1 register is updated by hardware, do not modify it by software. Please refer to *Central Aligned Mode (Count Up/Count)*.

Figure 8.56 shows an example of a center-aligned PWM waveform.

- PWM0_ARR=8
- PWM mode 1
- The flag bit is set in the following three cases (marked with an arrow)
  - Only when the counter counts down (CMS=01)
  - Only when the counter counts up (CMS=10)
  - When the counter counts up and down (CMS=11)

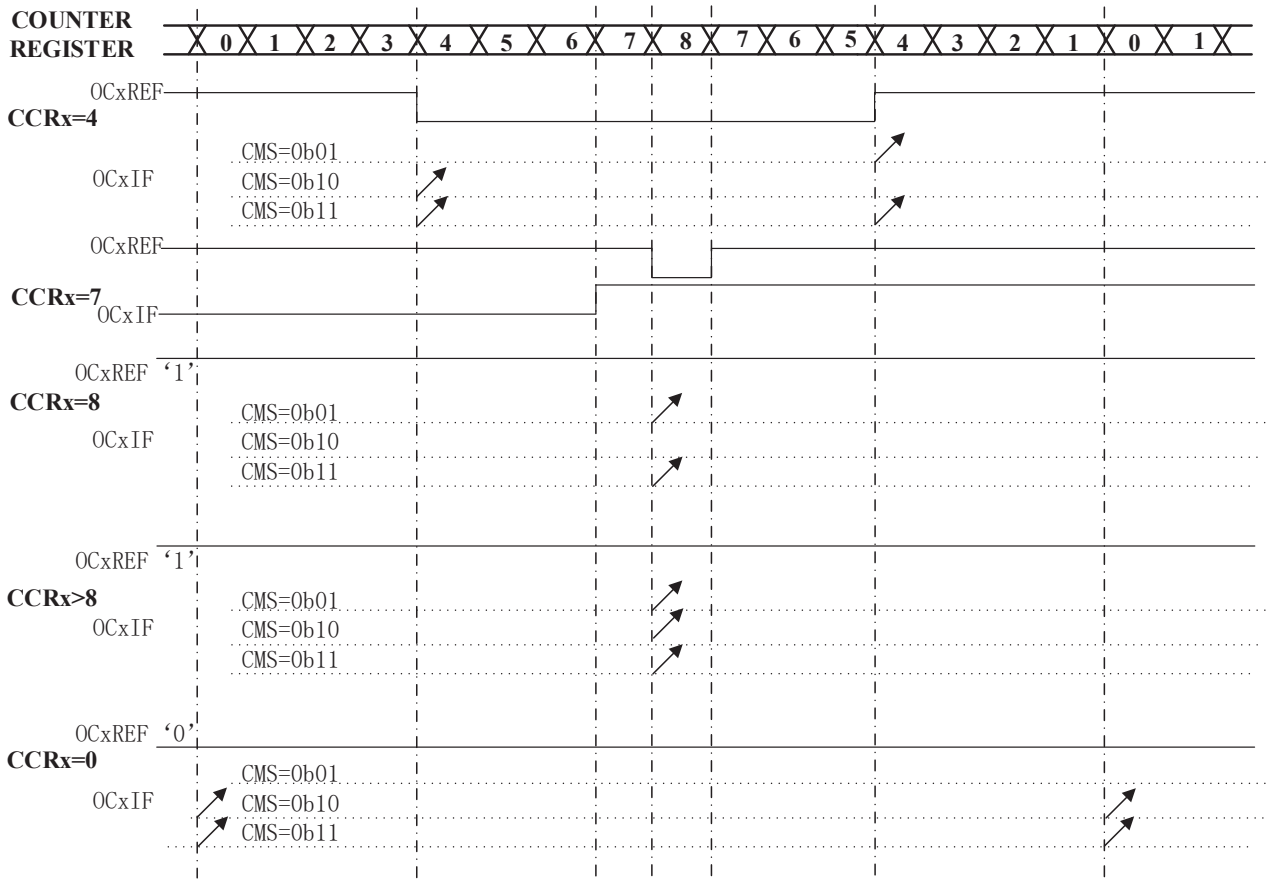


Figure 0.56 Center-aligned PWM waveform (ARR=8)

8.1.4.7.3 One Pulse Mode

The one pulse mode (OPM) is a special case of the many modes described earlier. This mode allows the counter to respond to an stimulus and to generate a pulse with a programmable length after a programmable delay. The counter can be started by the clock/trigger controller to generate waveforms in output compare mode or PWM mode. Setting the OPM bit in the PWM0_CR1 register will select the one pulse mode, when the counter automatically stops at the next UEV.

A pulse can be generated only if the comparison value is different from the initial counter value. Before starting (when the timer is waiting to be triggered), it must be configured as follows.

- up-counting mode: CNT < CCRx ≤ ARR.
- Down-counting mode: CNT > CCRx.

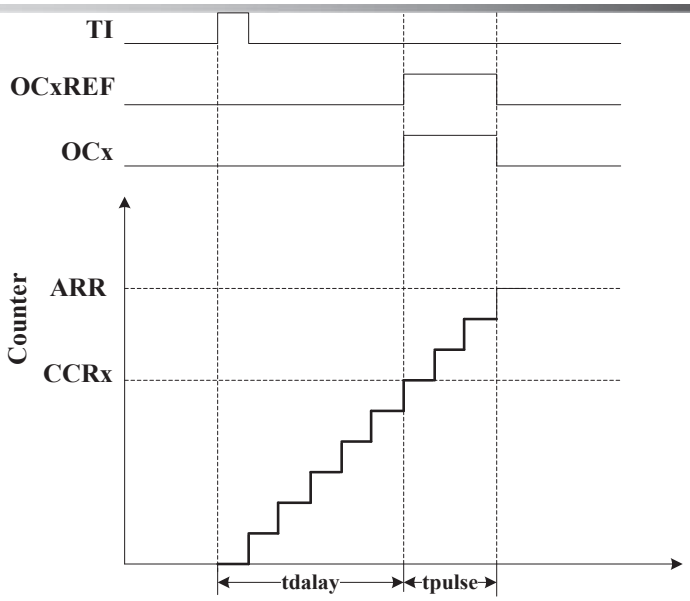


Figure 0.57 One Pulse Mode

For example, you need to delay t_{delay} after a rising edge is detected from the TI2 input pin to generate a positive pulse of t_{pulse} width on OC1.

It is assumed that IC2 is the trigger source for trigger channel 1:

- Write CC2S=01 of PWM0_CCMR2 register to map IC2 to TI2.
- Write CC2P=0 of PWM0_CCER1 register to enable IC2 to detect rising edge.
- Write TS=110 of PWM0_SMCR register to make IC2 as the trigger source of clock/trigger controller (TRGI).
- Write SMS=110 (trigger mode) in PWM0_SMCR register, IC2 is used to start the counter.
- The waveform of the OPM is determined by the value written to the comparison register (taking into account the clock frequency and the counter prescaler)
- t_{delay} is defined by the value in the PWM0_CCR1 register.
- The t_{pulse} is defined by the difference between the auto-reload value and the comparison value (PWM0_ARR-PWM0_CCR1).
- Assume that a waveform from 0 to 1 is to be generated when a comparison match occurs, and a waveform from 1 to 0 is to be generated when the counter reaches the preload value; first set OCxM=111 in the PWM0_CCMR1 register to enter PWM mode 2; selectively set OC1PE=1 in the PWM0_CCMR1 register as needed, set ARPE in the PWM0_CR1 register to enable the preload register; then fill in the comparison value in the PWM0_CCR1 register, fill in the auto-reload value in the PWM0_ARR register, and set the UG bit to generate an update event and wait for an external trigger event on TI2.

In this example, the DIR and CMS bits in the PWM0_CR1 register should be low.

Since only one pulse is needed, set OPM=1 in the PWM0_CR1 register to stop counting at the next update event (when the counter flips from the auto-reload value to 0).

Note: In one pulse mode, The CEN bit can be configured only after the UG bit is configured and a period time is delayed, because the UG bit will pull down the CEN bit in hardware after a period of

time.

8.1.4.7.4 Special case: OCx fast enable

In one pulse mode, edge detection of the TIx sets the CEN to 1, indicating that the counter is enabled. The output will toggle when a comparison between the counter value and the comparison value occurs. However, multiple clock cycles are required to complete these operations, which can limit the minimum possible delay (t_{delay} minimum).

If the waveform needs to be output with minimum delay, you can set OCxFE bit in the PWM0_CCMRx register to 1. This will force OCxRef (and OCx) to respond to the stimulus signal and no longer consider the result of the comparison. Its new level is the same as when the comparison match occurs. OCxFE only works when the channel is configured as PWM1 or PWM2 mode.

8.1.4.7.5 Complementary Output and Deadtime Insertion

PWM0 can output two complementary signals and managing the transient on and off of the outputs. Please refer to Figure 8.3.

This period of time is generally called deadtime. The user should adjust the deadtime according to the characteristics of the devices connected to the outputs (delay of level transition, delay of power switch, etc.). Please refer to Figure 8.46.

Configuring the CCxP and CCxNP bits in the PWM0_CCERx register allows you to select the polarity (main output OCx or OCxN) for each output independently.

The complementary signals OCx and OCxN are controlled by a combination of the following control bits: the CCxE and CCxNE bits in the PWM0_CCERx register, and the MOE, OISx, OISxN, OSSI, and OSSR bits in the PWM0_BKR register, as detailed in the following table.

Note: Deadtime Control is activated at the transition to the IDLE state (MOE drops to 0).

Control bit					Output status	
MOE bit	OSSI bit	OSSR bit	CCxE bit	CCxNE bit	Ocx output status	OCxNE output status
1	X	0	0	0	Output disabled (disconnected from timer)	Output disabled (disconnected from timer)
		0	0	1	Output disabled (disconnected from timer)	OCxREF + polarity OCx = OCxREF xor CCxNP
		0	1	0	OCxREF + polarity OCx = OCxREF xor CCxP	Output disabled (disconnected from timer)

		0	1	1	OCxREF + polarity + deadband	OCxREF Inverted + Polarity + Deadband
		1	0	0	Output disabled (disconnected from timer)	Output disabled (disconnected from timer)
		1	0	1	Off state (output enabled and disabled level) OCx = OCxP	OCxREF + polarity OCx = OCxREF xor CCxNP
		1	1	0	OCxREF + polarity OCx = OCxREF xor CCxP	Off state (output enabled and null level) OCxN = OCxNP
		1	1	1	OCxREF + polarity + deadband	OCxREF Inverted + Polarity + Deadband
0	0	X	X	X	Output disabled (disconnected from timer)	
	0					
	0					
	0					
	1				Off state (output enabled and inactive level) Asynchronously: OCx=CCxP, OCxN=CCxNP. Then, if the clock exists: after a dead time OCx=OISx and OCxN=OISiN, assuming that OISx and OISxN do not both correspond to active levels of OCx and OCxN.	
	1					
	1					
	1					

Figure 0.58 Complementary Output and Deadtime Insertion Table

Setting both the CCxE and CCxNE bits will insert a deadtime, and because of the break circuit, the MOE bit also needs to be set. There is an 8-bit deadtime generator for each channel. The reference signal OCxREF can generate 2 output channels OCx and OCxN. If OCx and OCxN are active high,

- The Ocx output signal is the same as the reference signal, except that its rising edge has a delay relative to the rising edge of the reference signal.
- The OcxN output signal is the opposite of the reference signal, except that its rising edge has a delay relative to the falling edge of the reference signal.

The following diagrams show the relationship between the output signal of the deadtime generator

and the present reference signal OCxREF. (Assuming CCxP=0, CCxNP=0, MOE=1, CCxE=1 and CCxNE=1)

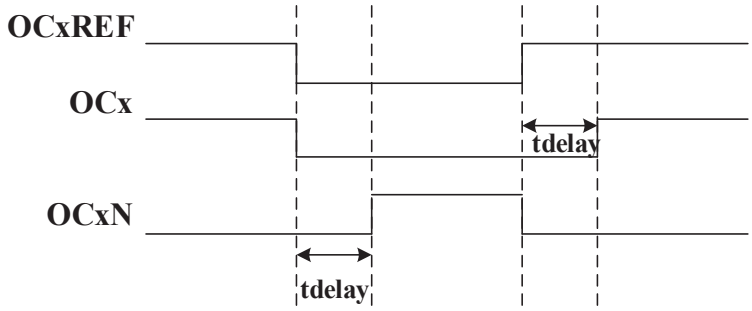


Figure 0.59 Complementary output with deadtime insertion

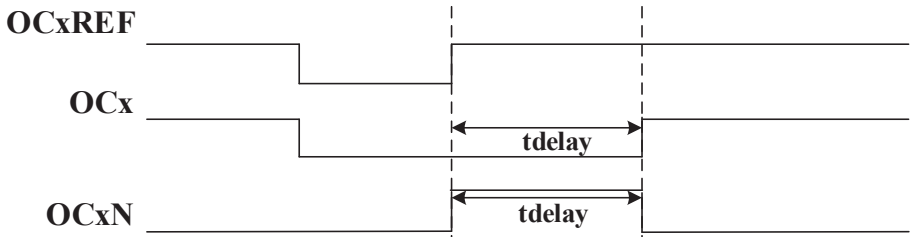


Figure 0.60 Deadtime waveform delay greater than negative pulse

The above figure shows that the deadtime delay t_{delay} is generated at the rising edge of OCxREF when the deadtime waveform delay is greater than the negative pulse.

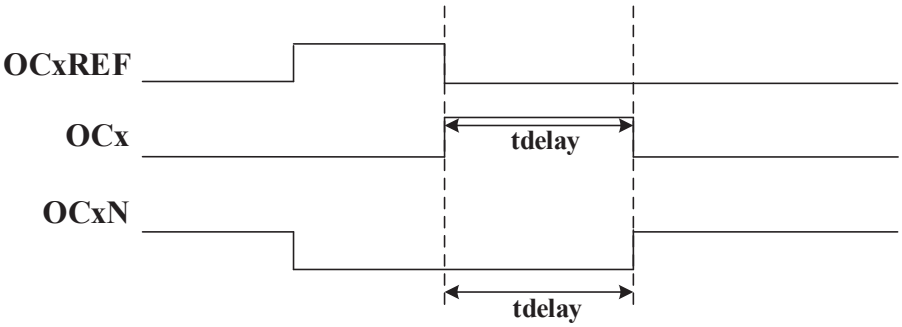


Figure 0.61 Deadtime waveform delay greater than positive pulse

The figure above shows that the deadtime delay t_{delay} is generated at the falling edge of OCxREF when the deadtime waveform delay is greater than the positive pulse.

Note:

- The dead time delay is the same for each channel and is programmed by the DTG bit in the PWM0_DTR register. For delay count, please refer to Deadtime Register (PWM0_DTR).
- When the delay is greater than the present active output width, the output PWM waveform will

have an error.

8.1.4.7.6 Redirect OCxREF to OCx or OCxN

In output mode (forced set, output compare or PWM), OCxREF can be redirected to the output of OCx or OCxN by configuring the CCxE and CCxNE bits of the PWM0_CCERx register.
This function can send a special waveform (e.g. PWM or static active level) on one of the outputs when the complementary output is at a inactive level. Another function is to have both outputs at inactive level or at active level at the same time (when it is still a complementary output with deadtime).

Note:
When only OCxN is enabled (CCxE=0, CCxNE=1), it does not invert the phase and is active immediately when OCxREF becomes high. For example, if CCxNP=0, then OCxN=OCxREF. On the other hand, when both OCx and OCxN are enabled (CCxE=CCxNE=1), OCx is active when OCxREF is high; however it is quite opposite for OCxN, OCxN becomes active when OCxREF is low.

8.1.4.7.7 Six-step PWM output for motor control

When complementary outputs are required on a channel, the preload bits are OCxM, CCxE and CCxNE. These preload bits are passed to the shadow register bits when a COM phase change event occurs. This will preset the next configuration and modify the configurations of all channels simultaneously at the same moment. COM can be generated via software by setting the COMG bit in the PWM0_EGR register or via hardware on the rising edge of TRGI.
Figure 8.62 shows the OCx and OCxN outputs in three different configurations when a COM event occurs.

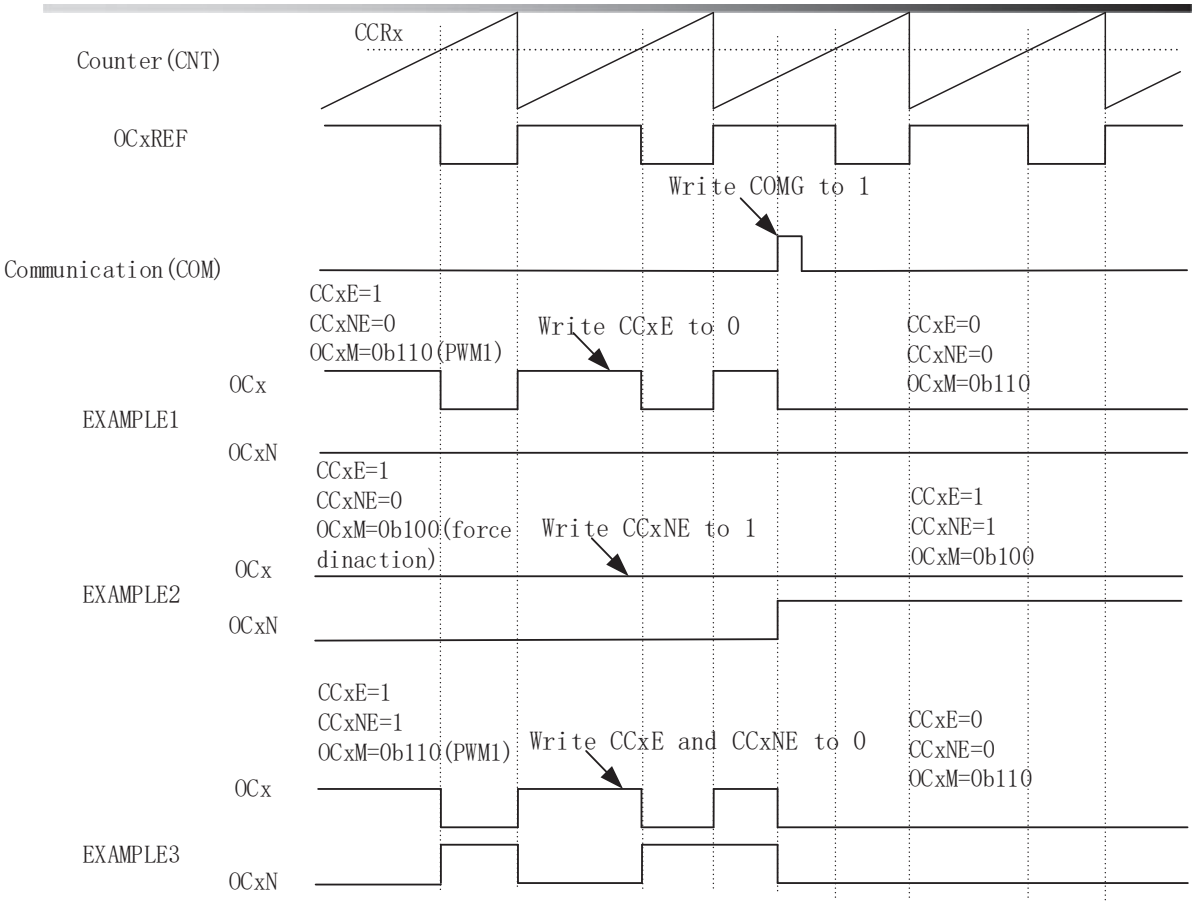


Figure 0.62 Example of generating a six-step PWM by COM (OSSR=1)

8.1.4.8 Break

Break are commonly used in motor control. When the break function is used, the output enable signal and inactive level are modified according to the corresponding control bits (MOE, OSSI and OSSR bits in the PWM0_BKR register).

After system reset, the break circuit is disabled and the MOE bit is low. Setting the BKE bit in the PWM0_BKR register enables the break function. The polarity of the break input signal can be selected by configuring the BKP bit in the same register. BKE and BKP bits can be modified at the same time.

The MOE falling edge can be asynchronous with respect to the clock module, so a resynchronization circuit is set up between the actual signal (acting on the output) and the synchronization control bit (in the PWM0_BKR register). This resynchronization circuit creates a delay between the asynchronous signal and the synchronous signal. In particular, if write MOE=1 when it is low, a delay (null instruction) must be inserted before reading it out to get the correct value. This is because the write is an asynchronous signal and the read is a synchronous signal.

When break occurs (selected level occurs at the break input),

- The MOE bit is cleared asynchronously, setting the output in either an invalid state, an idle state, or a reset state (as selected by OSSI). This feature remains in effect when the MCU

oscillator is off.

- Once MOE=0, each output channel outputs the level set by the OISx bit of the PWM0_OISR register. If OSSI=0, the timer no longer controls the output enable signal, otherwise the output enable signal is always high.
- When using complementary output,
  - The output is first set in a reset state, that is, an invalid state (depending on polarity). This is an asynchronous operation and is valid even when the timer is not clocked.
  - If the timer clock is still present, the deadtime generator will be re-enabled after the deadtime according to the levels of OISx and OISxN. Note: The dead time is a bit longer than usual because of the resynchronized MOE.
- If the BIE bit of PWM0_IER register is set, an interrupt is generated when the break status flag (BIF bit in PWM0_SR1 register) is '1'.
- If the AOE bit in the PWM0_BKR register is set, the MOE bit is automatically set on the next UEV. This can be used for waveform control, for example. Otherwise, the MOE always remains low until it is '1' again. This feature can be used for safety purposes, where you can connect the break input to a power-driven alarm output, thermal sensor, or other safety device.

Note:

The break inputs act on signal level. Therefore, when the break input is active, the MOE cannot be set at the same time (automatically or by software), and the status flag BIF cannot be cleared.

The break is generated by the BRK input (BKIN), which has a programmable active polarity and is enabled or disabled by the BKE bit of the PWM0_BKR register.

In addition to the break input and output management, the break circuit implements write protection to secure the application. It allows the user to freeze several configuration parameters (OCx polarity and state when disabled, OCxM configuration, break enable and polarity). The user can select one of the three levels of protection via the LOCK bit in the PWM0_BKR register. The LOCK bit field can only be modified once after MCU reset. Figure 8.63 shows an example of break response output.

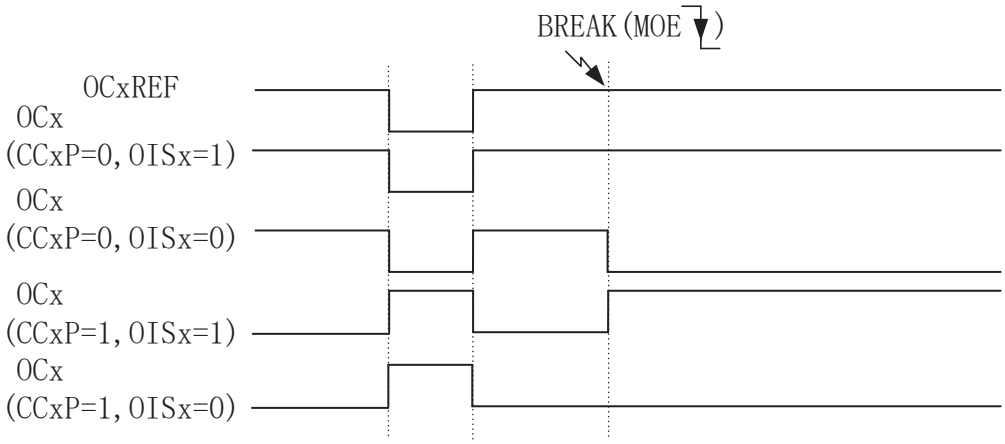


Figure 0.63 Break Response Output without Complementary Output Channel

Figure 8.64 shows an example of the break response with complementary output.

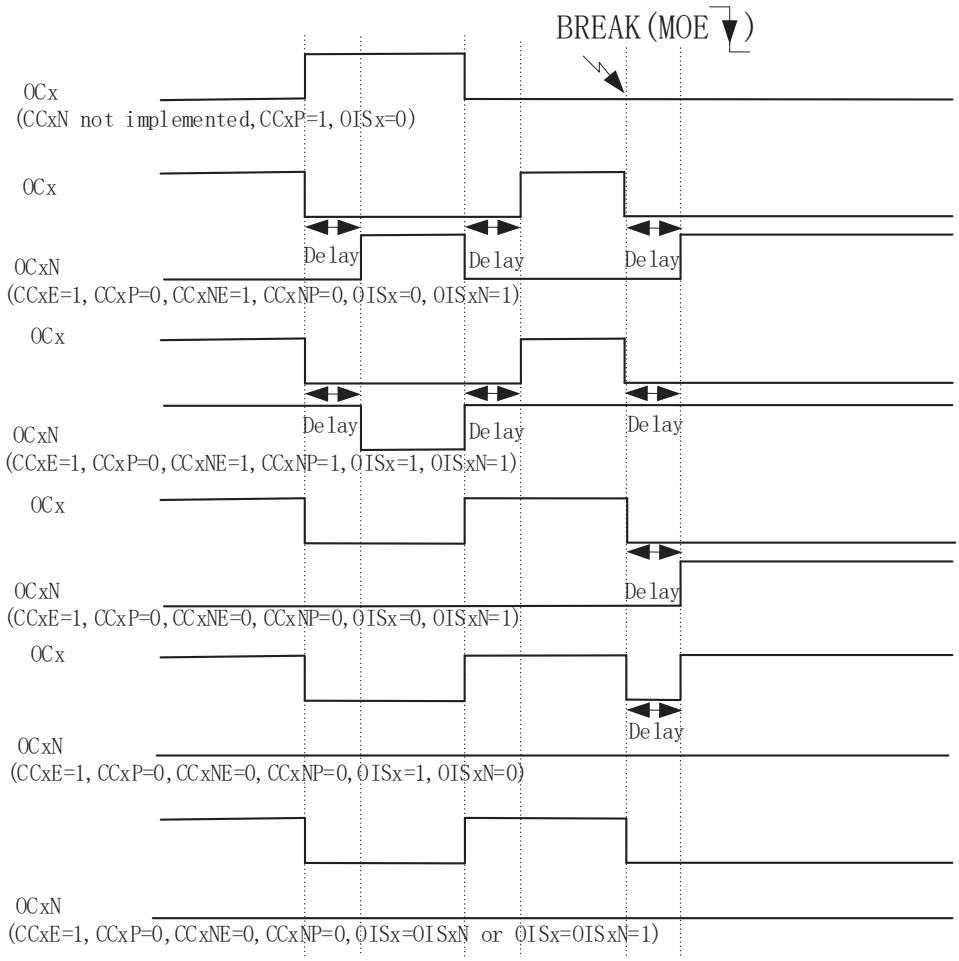


Figure 0.64 Output of break response (PWM0 complementary output)

8.1.4.9 Clear OCREF Signal on External Event

For a given channel, a high level at the ETRF input (setting the corresponding OCxCE bit in the PWM0_CCMRx register to 1) can pull the OCxREF signal low, and the OCxREF signal will remain low until the next UEV. This function can only be used in output compare and PWM mode and not in forced mode.

For example, the OCxREF signal can be linked to the output of a comparator for current control. In this case, the ETR must be configured as follows.

1) The external trigger prescaler must be off: ETPS[1:0] in the PWM0_ETR register = 00.
2) External clock mode 2 must be disabled: ECE=0 in PWM0_ETR register.
3) The external trigger polarity (ETP) and external trigger filter (ETF) can be configured as required.

Please refer to Figure 8.22 External trigger input block diagram.

The following figures shows the behavior of the OCxREF signal when the ETRF input becomes high, corresponding to different OCxCE values. In this example, timer PWM0 is set as PWM mode.

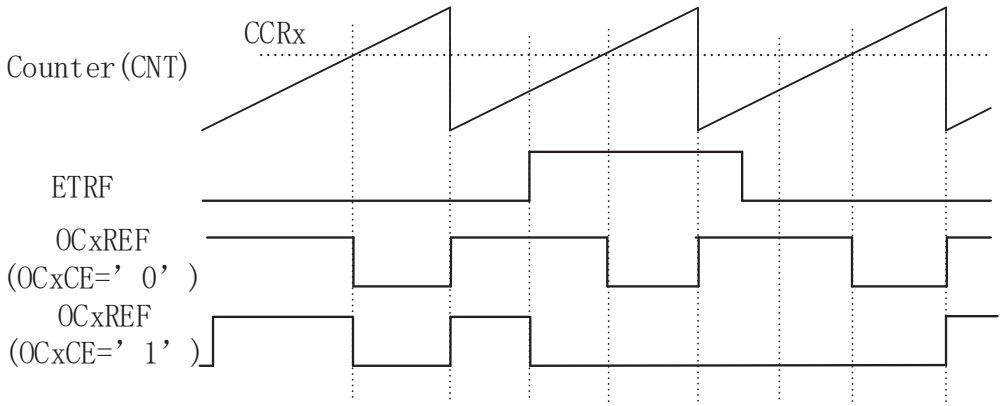


Figure 0.65 ETR clears OCxREF of PWM0

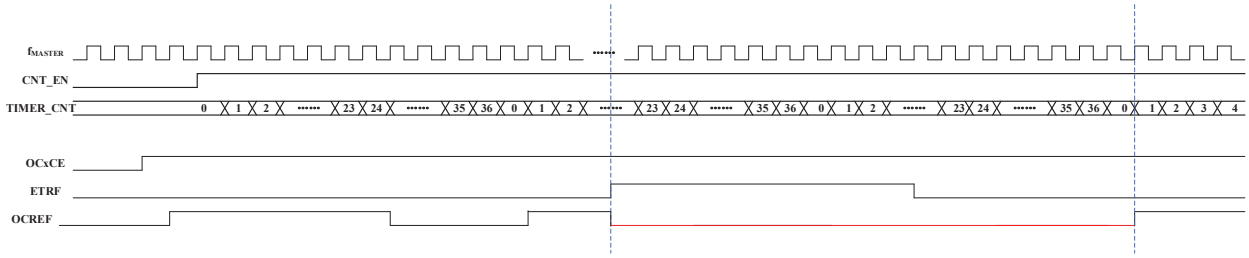


Figure 0.66 Timing Diagram that ETR clears OCxREF of PWM0

8.1.4.10 Encoder Interface Mode

This mode is generally used for motor control. The encoder interface mode is selected by setting SMS=001 in the PWM0_SMCR register if the counter counts only on the edge of TI2, by setting SMS=010 if it counts only on the edge of TI1, or by setting SMS=011 if the counter counts on both the TI1 and TI2 edges.

Setting the CC1P and CC2P bits in the PWM0_CCER1 register allows you to select TI1 and TI2 polarities; you can also program the input filter if necessary.

The two inputs TI1 and TI2 are used as interfaces to the incremental encoder (Table 0.67 Relationship between Count Direction and Encoder Signal

). Assume that the counter has been started (CEN=1 in the PWM0_CR1 register), the counter counts each time a valid jump is generated on TI1FP1 or TI2FP2. TI1FP1 and TI2FP2 are the signals of TI1 and TI2 after passing through the input filter and polarity control. Without filter and polarity transformation, TI1FP1 = TI1, TI2FP2=TI2. The counting pulse and direction signal are generated according to the jump order of the two input signals. The counter counts up or down based on the

jump order of the two input signals, while the hardware sets the DIR bit of the PWM0_CR1 register accordingly. Regardless of whether the counter counts on TI1, on TI2, or on both TI1 and TI2, the DIR bit is recalculated on the jump edge of either input (TI1 or TI2).

The encoder interface mode is basically similar to an external clock with direction selection. This means that the counter only counts continuously between 0 and the auto-reload value of the PWM0_ARR register (either 0 to ARR or ARR to 0, depending on the direction). So PWM0_ARR must be configured before counting can start; again, the catcher, comparator, prescaler, repetition counter, trigger output characteristics still work normally.

Note: Encoder mode and external clock mode 2 are not compatible and therefore cannot be operated simultaneously.

In this mode, the counter is automatically modified according to the speed and direction of the incremental encoder, so that the content of the counter always points to the position of the encoder. The counting direction corresponds to the direction of rotation of the connected sensor. Table 8.67 shows all possible combinations, assuming that TI1 and TI2 do not change at the same time.

Active edge	Level of corresponding signal (TI1FP1 for TI2, TI2FP2 for TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Only count at TI1	High	Count down	Count up	No Count	No Count
	Low	Count up	Count down	No Count	No Count
Only count at TI2	High	No Count	No Count	Count up	Count down
	Low	No Count	No Count	Count down	Count up
Count on TI1 and TI2	High	Count down	Count up	Count up	Count down
	Low	Count up	Count down	Count down	Count up

Table 0.67 Relationship between Count Direction and Encoder Signal

An external incremental encoder can be connected directly to the MCU without the need for external interface logic. However, a comparator is generally used to convert the differential output of the encoder into a digital signal, which greatly increases the noise immunity. The third signal from the encoder outputs represents the mechanical zero point, which can be connected to an external

interrupt input and triggers a counter reset.

Figure 8.68 is an example of counter operation, showing the generation and direction control of the count signal. It also shows how input jitter is compensated where both edges are selected. This might occur if the sensor is positioned near one of the switching points. In this example, we assume the configuration as follows:

- CC1S='01' (PWM0_CCMR1 register, IC1FP1 mapped to TI1)
- CC2S='01' (PWM0_CCMR2 register, IC2FP2 mapped to TI2)
- CC1P='0' (PWM0_CCER1 register, IC1 is not inverted, IC1=TI1)
- CC2P='0' (PWM0_CCER1 register, IC2 is not inverted, IC2=TI2)
- SMS='011' (PWM0_SMCR register, all inputs are valid on rising and falling edges)
- CEN='1' (PWM0_CR1 register, counter enabled)

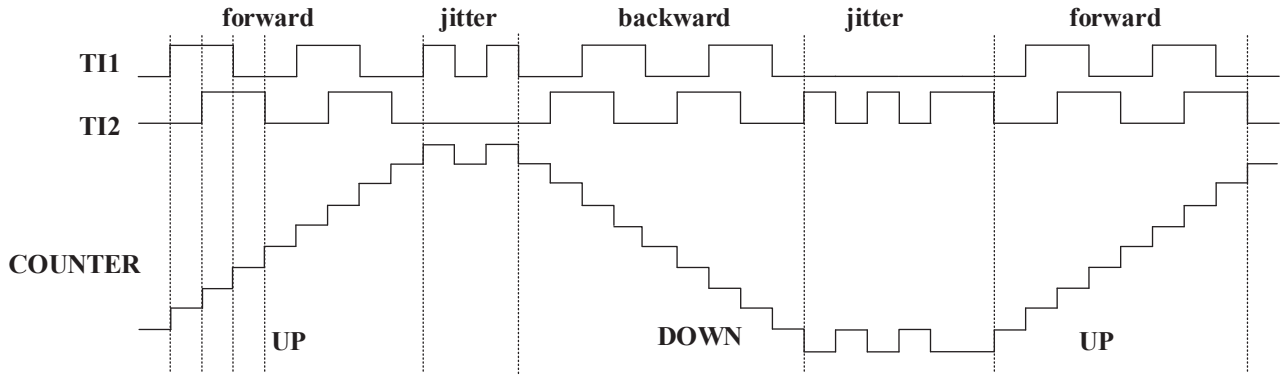


Figure 0.68 Example of counter operation in encoder mode

Figure 8.69 shows an example of counter operation when IC1 polarity is reversed (CC1P='1', other configurations are the same as the example above).

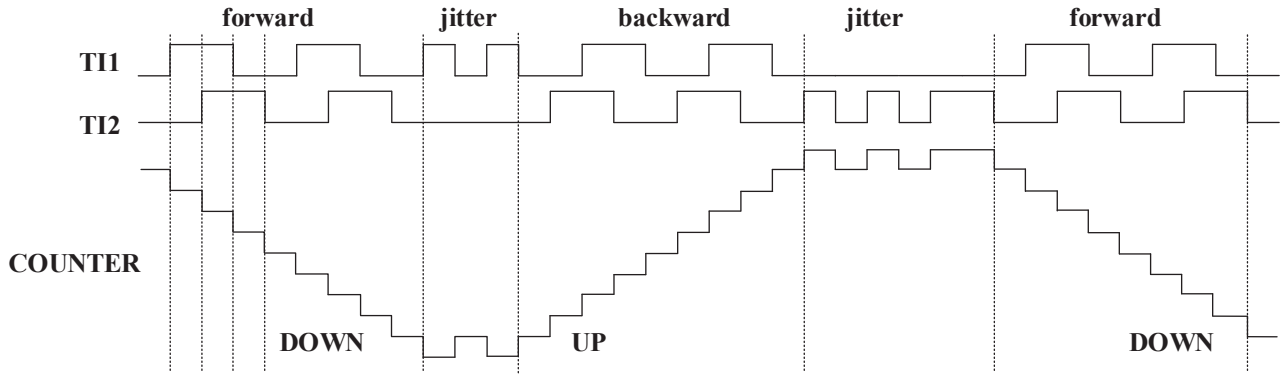


Figure 0.69 Example of encoder interface mode with IC1 inverted

When the timer is configured in encoder interface mode, information about the current position of the sensor is provided. Information on the dynamics (velocity, acceleration, deceleration) can be obtained by measuring the interval between two encoder events using another timer configured in capture mode. The encoder output indicating the mechanical zero point can be used for this purpose. Depending on the interval between the two events, the counter can be read out at a certain time interval. If possible, you can latch the counter value to a third input capture register (the capture

signal must be periodic and can be generated by another timer).

8.1.5 Interrupt

PWM0 has 8 interrupt request sources.

- Break interrupt
- Trigger interrupt
- COM event interrupt
- Input capture/output compare interrupt 4
- Input capture/output compare interrupt 3
- Input capture/output compare interrupt 2
- Input capture/output compare interrupt 1
- Update event interrupts (e.g., counter overflow, underflow and initialization)

To use the interrupt feature, set the corresponding interrupt enable bits in the PWM0_IER register for each interrupt channel being used: i.e. BIE, TIE, COMIE, CCxIE, UIE bits.
Each of the above interrupt sources can also be generated via software by setting the corresponding bit in the PWM0_EGR register.

8.2 16-bit General Purpose Timer PWM1

The general-purpose timer consists of a 16-bit auto-reload counter with a programmable prescaler. The timer can be driven by an external crystal or PLL_24MHz.

It is suitable for a variety of applications, including.

- Basic Timing
- Measure the pulse length of the input signal (input capture)
- Generate output waveforms (output comparison, PWM and single pulse)
- Sync with other timers or external signals (external clock, reset, trigger and enable signals)

This chapter only introduces the main functions of the general-purpose timer, for more information on the functions, please refer to *8.1 16-bit Advanced Timer* PWM0 .

8.2.1 Main Functions

The main functions of PWM1 are as follows.

- 16-bit auto-reload up counter.
- 4-bit programmable(Support real-time modification) prescaler allowing the counter clock frequency to be divided by any power of 2 from 1 to 32768.Synchronous circuit for timer control by external signals and timer interconnection.
- The 3 independent channels can be configured to.
  - Input capture.

- Output comparison.
- PWM generation (edge aligned mode).
- One pulse mode output.

- Events that generate interruptions include:
  - Update: counter overflow, counter initialization (via software).
  - Trigger events
  - Input capture.
  - Output compare.

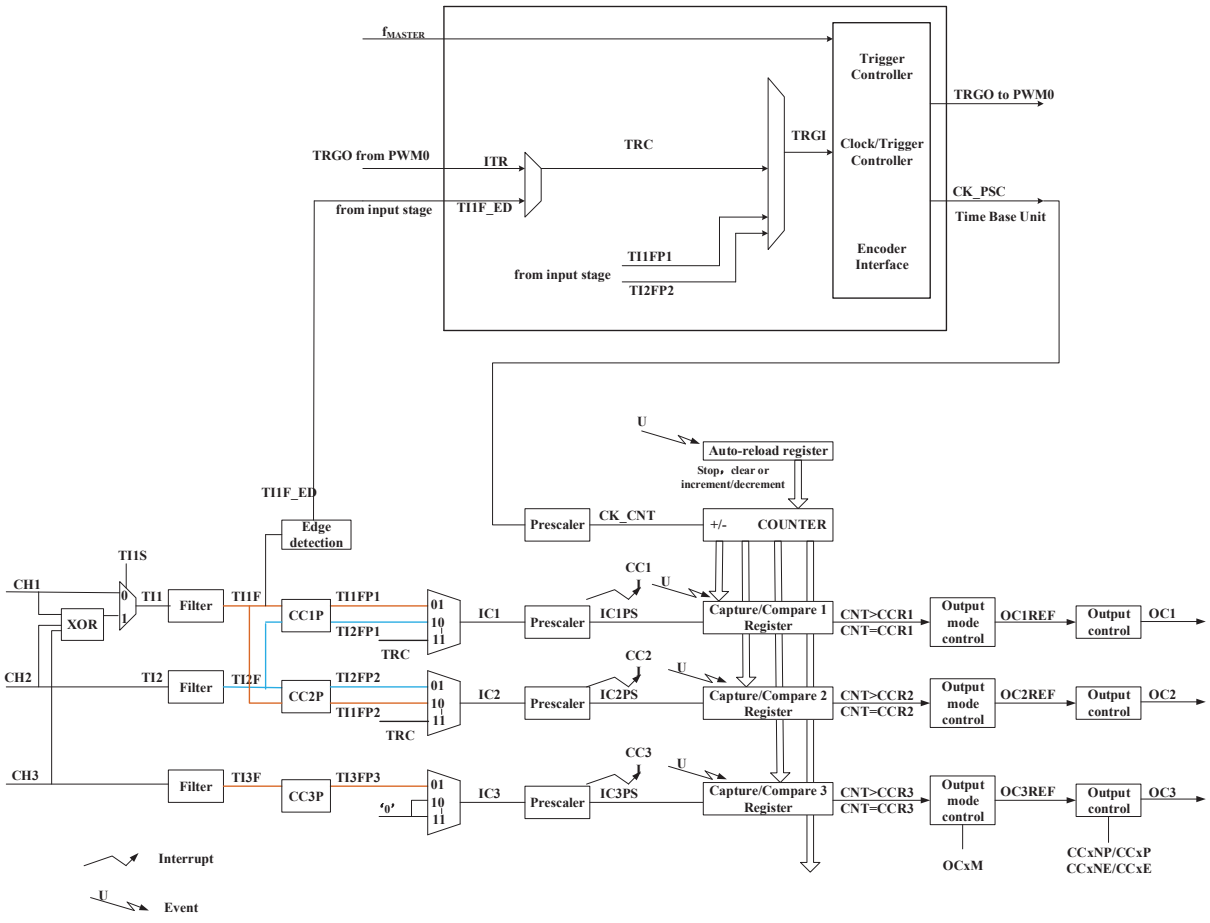


Figure 0.70 PWM1 General Purpose Timer Block Diagram

8.2.2 Time Base Unit

The Time Base Unit includes a 16-bit up counter, a prescaler and a 16-bit auto-reload register. It has no repetition counter. The counter uses an internal clock (f_{MASTER}), which is provided by CK_PSC and divided by a prescaler to generate the counter clock CK_CNT.

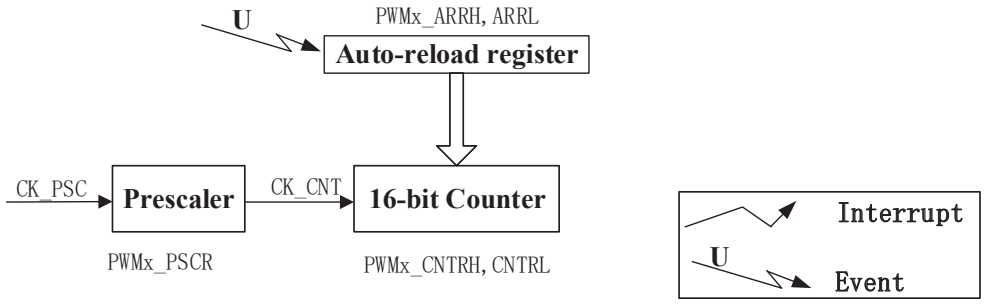


Figure 0.71 Time Base Unit

For more information, please refer to PWM0 Time Base Unit.

8.2.2.1 Prescaler

The prescaler for PWM1 is based on a 16-bit counter controlled by a 4-bit register (PWM1_PSCR). Since the control register has a buffer, it can be changed at runtime. The counter frequency can be divided by any power of 2 from 1 to 32768. The clock frequency of the counter can be calculated as follows:

$$f_{CK\_CNT} = f_{CK\_PSC}/2^{(PSCR[3:0])}$$

The value of the prescaler is written by the preload register. Once the low 8 bits of the preload register are written, the shadow register with the current value to be used is written with the new value.
The prescaler value requires two separate write operations to write the 16-bit register, with the high bit written first. Do not use the LDW instruction to write the low bit first.
The new prescaler value is adopted when the next update event occurs.
The read operation of PWM1_PSCR register is via the preload register, so the read is not restricted.

8.2.2.2 Read/Write 16-bit Counter

The operation of writing counter has no cache and can write the PWM1_CNTRH and PWM1_CNTRL registers at any time. Therefore, it is recommended not to write a new value while the counter is running to avoid writing the wrong value.
The operation of reading counter comes with an 8-bit cache. After the high byte is read, the low byte is automatically cached and the cached data does not change until the 16-bit read operation is completed.
Note: Do not read the lower byte first, as the value read out this way is wrong.
For more information, please refer to PWM0 *8.1.2.1 Read/Write 16-bit* Counter.

8.2.2.3 Read/Write 16-bit PWM1_ARR

The value of the preload register will be written to the 16-bit PWM1_ARR register, with the high byte

written first and the low byte next. The shadow register is locked when the high byte is written and held until the low byte is written.

Note: Do not write the low byte first, as the value written this way is wrong.

For more information, please refer to the PWM0*8.1.2.2* .

8.2.2.4 up-counting mode

In up-counting mode, the counter counts from 0 to a user-defined comparison value (the value of the PWM1_ARR register), then restarts counting and generates an overflow event. Also, if the UDIS bit of the PWM1_CR1 register is 0, an update event will be generated which will update the flag bit UIF. If the update interrupt enabled UIE=1, an update interrupt will be generated.

Setting the UG bit of the PWM1_EGR register (either by software or using a slave mode controller) can also generate an update event.

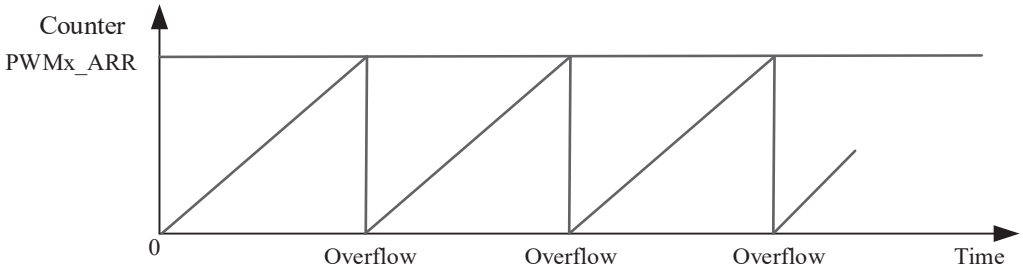


Figure 0.72 up-counting mode

Using software to set the UDIS bit in the PWM1_CR1 register disables the update event, but the counter is still cleared to 0 and the prescaler is also cleared to 0. In the case of UDIS=0, if URS=1 is set, setting the UG bit at this time will generate an update event, but the hardware does not set the UIF flag (i.e., no interrupt request is generated). This is to avoid generating both update and capture interrupts when the counter is cleared in capture mode.

When an update event occurs, all registers will be updated and (depending on the setting of the URS bit) the update flag bit (UIF bit of the PWM1_SR register) will also be set.

(1) If the ARPE bit of the PWM1_CR1 register is 1, the auto-reload shadow register is updated with the value of the preload register (PWM1_ARR).

2) The buffer of the prescaler is reloaded with the value of the preload register (PWM1_PSCR).

Note:

The counter starts counting only after the CEN bit is enabled and one clock cycle (CK_CNT) has passed.

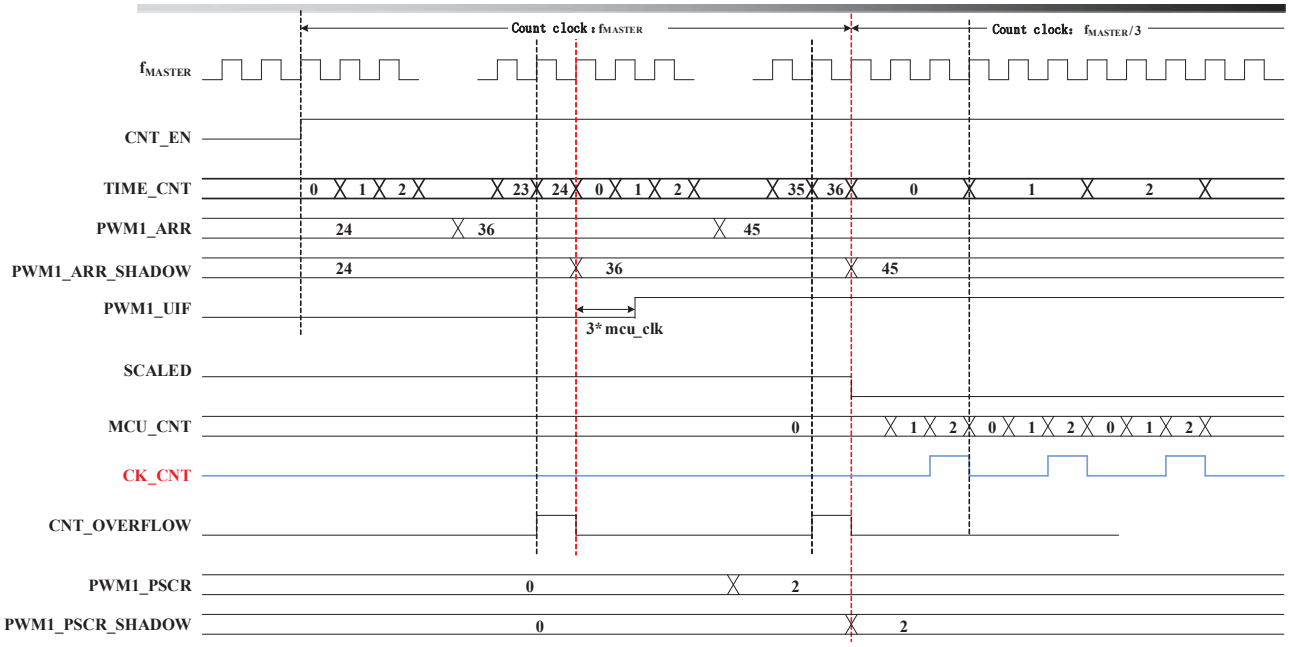


Figure 0.73 Count up Timing Sequence

For more information, please refer to PWM0 up-counting mode.

8.2.3 Clock/Trigger Controller

The clock/trigger controller allows the user to select the clock source, input trigger signal and output signal for the counter. See Figure 8.74.

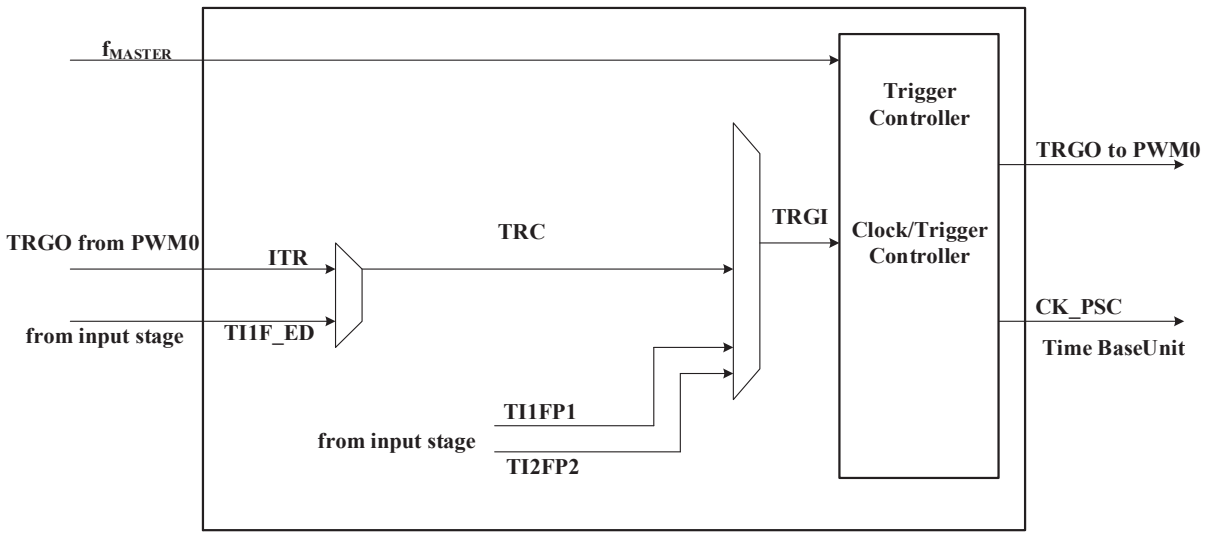


Figure 0.74 Clock/Trigger Controller Block Diagram

For more information, please refer to PWM0 Clock/Trigger Controller.

8.2.3.1 Prescaler (CK_PSC)

The prescaled clock (CK_PSC) of Time Base Unit can be provided by the following sources.

- Internal clock (f_{MASTER})
- External clock mode 1: External clock input (TIx)
- Internal Trigger Input (ITR): Use the advanced control timer PWM0 as the prescaled clock for the general-purpose timer PWM1.

8.2.3.2 Internal Clock Source (f_{MASTER})

If both the trigger mode controller and the external trigger input are disabled (SMS=000 in the PWM1_SMCR register), the CEN, DIR and UG bits are effectively control bits and can only be modified by software (the UG bit is automatically cleared). Once the CEN bit is set to 1, the prescaler is clocked by the internal clock f_{MASTER}.
For more information, please refer to PWM0 Internal Clock Source (f_{MASTER}).

8.2.3.3 External Clock Source Mode 1

This mode is selected when SMS=111 in the PWM1_SMCR register. The counter can count on each rising or falling edge of the selected input. When a rising edge occurs at the selected input, the counter increases by 1 and the trigger flag bit (TIF bit in PWM1_SR1 register) is set to 1. If an interrupt is enabled (configured in PWM1_IER register), an interrupt request is generated.

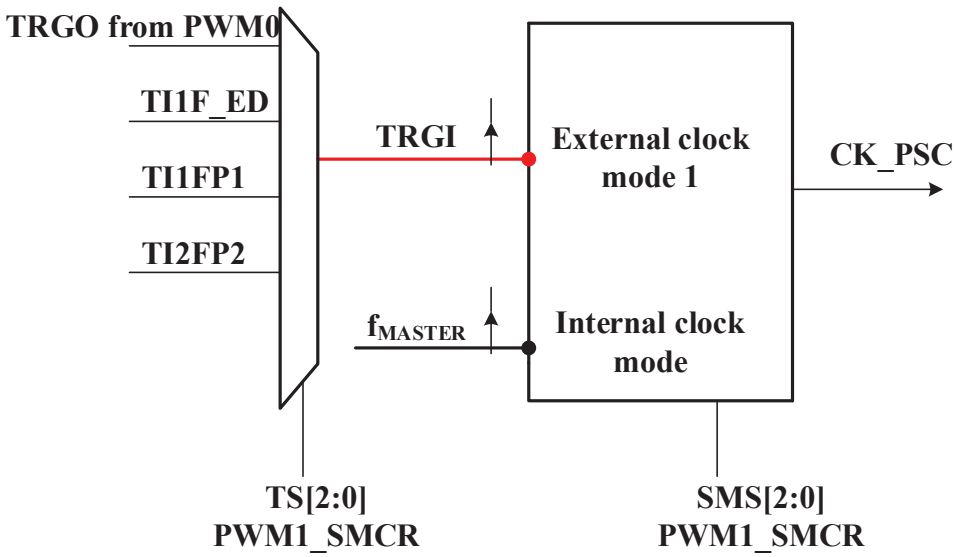


Figure 0.75 External Clock Source Mode 1 Block Diagram

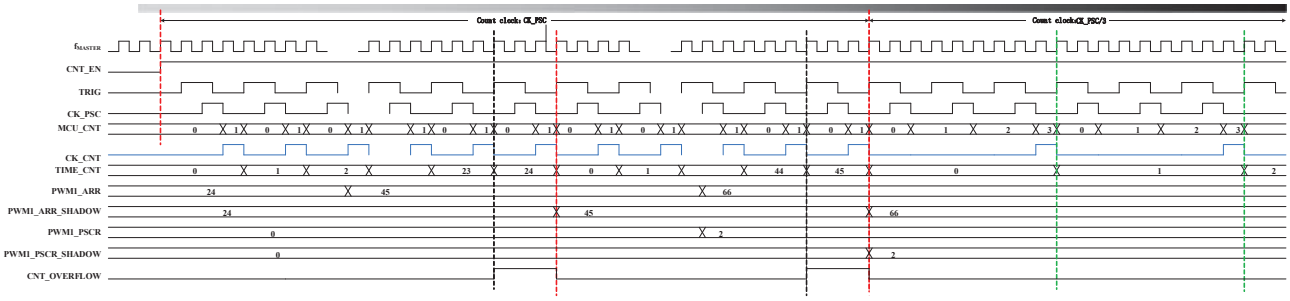


Figure 0.76 External Clock Source Mode 1 Count Up Timing Diagram

For more information, please refer to the PWM0 External Clock Source Mode 1.

8.2.3.4 Sync Trigger

The counter allows three trigger inputs.

- TI
- T2
- TRGO from PWM0

The PWM1 counter uses three modes to synchronize with the external trigger signal: standard trigger mode, reset trigger mode, and gated trigger mode.
For more information, please refer to PWM0 Trigger Synchronization.

8.2.3.4.1 Standard Trigger Mode

This mode is selected when SMS=110 of PWM1_SMCR register. The counter is started on the rising edge of the trigger input TRGI, and only the start of counter is controlled. The counter is enabled depending on events on the selected input (at this point the count enablement is independent of the CEN bit in the PWM1_CR1 register).
When there is a rising edge on TRGI, the counter starts counting driven by the internal clock f_{MASTER}, and set the TIF flag bit. The delay between the rising edge of the TRGI and the start of the counter depends on the resynchronization circuit at the TRGI input.

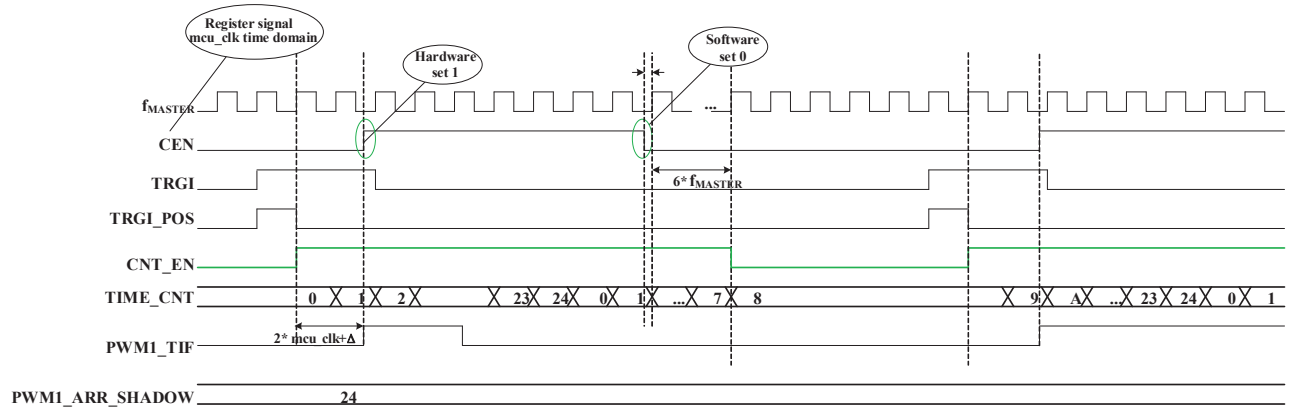


Figure 0.77 Standard Trigger Timing Diagram

8.2.3.4.2 Reset Trigger Mode

This mode is selected when SMS = 100 in the PWM1_SMCR register. When a trigger input event occurs, the counter and its prescaler can be reinitialized; at the same time, an UEV is also generated if the URS bit of the PWM1_CR1 register is low; then all the preload registers (PWM1_ARR, PWM1_PSCR, PWM1_CCRx) are updated.
Set CEN=1 in the PWM1_CR1 register, the counter starts counting according to the internal clock f_{MASTER} until a rising edge appears at TRGI, when the counter is initialized and starts counting again. At the same time, the trigger flag (TIF bit of PWM1_SR1 register) is set and an interrupt request is generated if an interrupt is enabled (TIE bit of PWM1_IER register). The delay between the rising edge of the TRGI and the actual reset of the counter depends on the resynchronization circuit at the TRGI input.

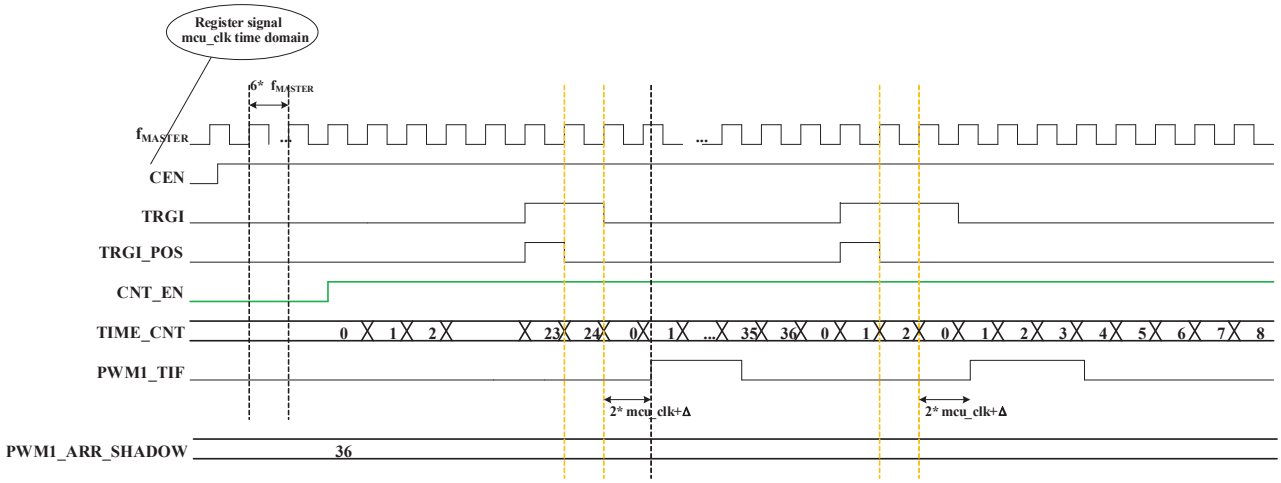


Figure 0.78 Reset Trigger Timing Sequence

8.2.3.4.3 Gated trigger mode

This mode is selected when SMS=101 of PWM1_SMCR register. The counter is enabled by the TRGI level of the selected input signal. In the gated mode, if CEN=0, the counter cannot start, regardless of the trigger input level. As long as the TRGI level is valid, the counter starts counting based on the internal clockf_{MASTER} and stops counting once the TRGI level is invalid. The TIF flag bit will be set when the counter starts or stops counting. The delay between the active edge of TRGI and the actual stop of the counter depends on the resynchronization circuit at the TRGI input.

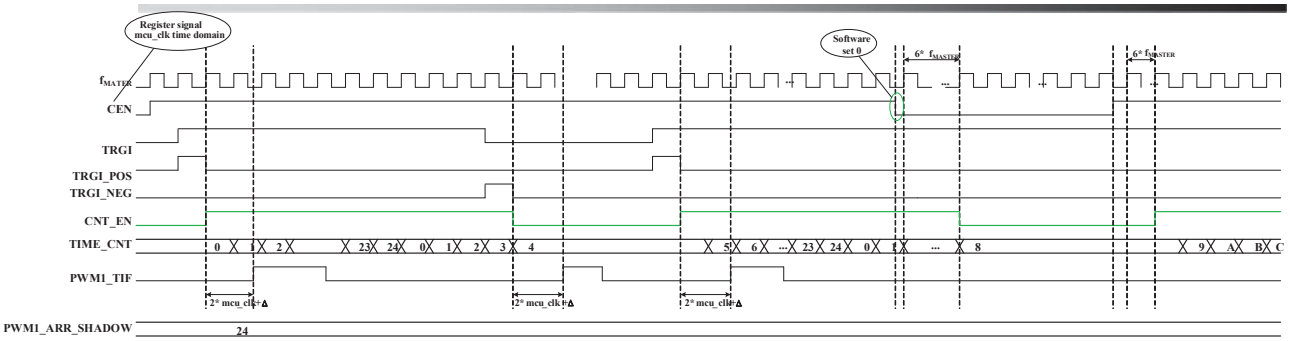


Figure 0.79 Gated Trigger Timing Diagram

8.2.3.5 Sync with Advanced Timer PWM0

Timers are interconnected inside the chip for synchronization or linking. When a timer is configured as master mode, it can output a trigger signal (TRGO) to those timers configured as slave mode to complete reset, start, stop operations, or be a driving clock for those timers.

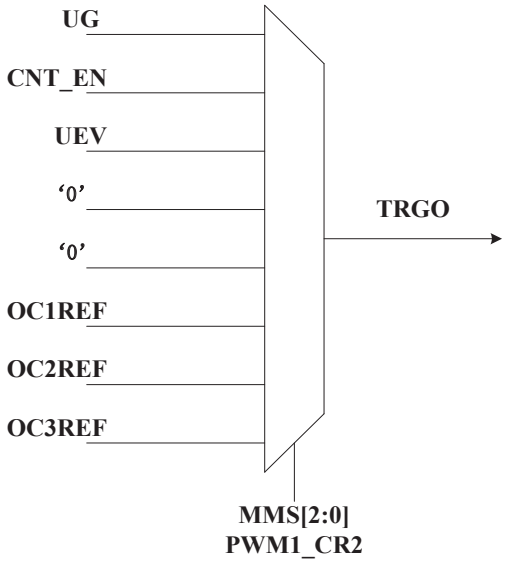


Figure 0.80 Trigger Master/Slave Mode Selection

8.2.4 Capture/Compare Channels

8.2.4.1 Read/Write 16-bit PWM1_CCRx Register

When the channel is configured as output mode, the writing operation of the 16-bit PWM1_CCRx register is completed via the preload register. The high byte must be written first. While writing the high byte, update to the shadow register is disabled until the low byte writing operation is completed, which is similar to writing operation of PWM1_ARR register. The read operation to the PWM1_CCRx register is completed via the preload register, so special attention is not needed.

When the channel is configured as input mode, the PWM1_CCRx register is read-only at this time. The read operation of the PWM1_CCRx register is similar to the read operation of the counter. When a capture occurs, the contents of the counter are captured into the PWM1_CCRx shadow register and later copied into the preload register.
The preload register is frozen while the read operation is in progress.

8.2.4.2 Input Capture Mode

Figure 8.81 shows a timer with two input channels, channel 1 is internally linked to the comparator.

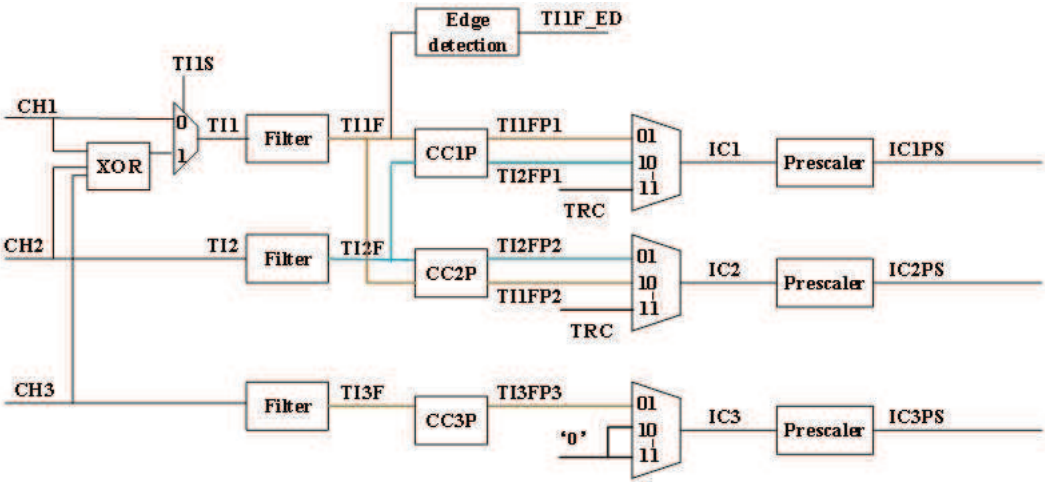


Figure 0.81 Inputs Diagram

In input capture mode, the present value of the counter is latched into the capture/compare register (PWM1_CCRx) when the corresponding edge on the ICx signal is detected. When a capture event occurs, the corresponding CCxIF flag (PWM1_SR register) is set to 1. If the CCxIE bit of the PWM1_IER register is set, that is, interrupts are enabled, an interrupt request will be generated. If the CCxIF flag is already high when a capture event occurs, the repetition capture flag CCxOF (PWM1_SR2 register) is set to 1. CCxIF is cleared by writing CCxIF=0 or by reading the capture data stored in the PWM1_CCRxL register. CCxOF is cleared by writing CCxOF=0.

When an input capture occurs,

- When a active level transition is generated, the value of the counter is passed to the PWM1_CCR1 register.
- The CCxIF flag is set (interrupt flag). CCxOF is also set to 1 when at least 2 consecutive captures occur and CCxIF has not been cleared.

- If the CCxIE bit is set, an interrupt will be generated.

Note: To handle overcapture (CCxOF bit), it is recommended to read the data before reading the repetition capture flag, which is to avoid losing repetition capture information that may be generated after reading the overcapture flag and before reading the data.

Note 2: Setting the corresponding CCxG bit in the PWM1_EGR register allows you to generate an input capture interrupt by software.

Note 3: When configuring the PWM1_CCMRx register, it is configured in two steps: first, configure the CCxS bit in the PWM1_CCMRx register to select the input mode mapping; second, configure the ICxF and ICxPSC in the PWM1_CCMRx register to select the filter and prescaler coefficients.

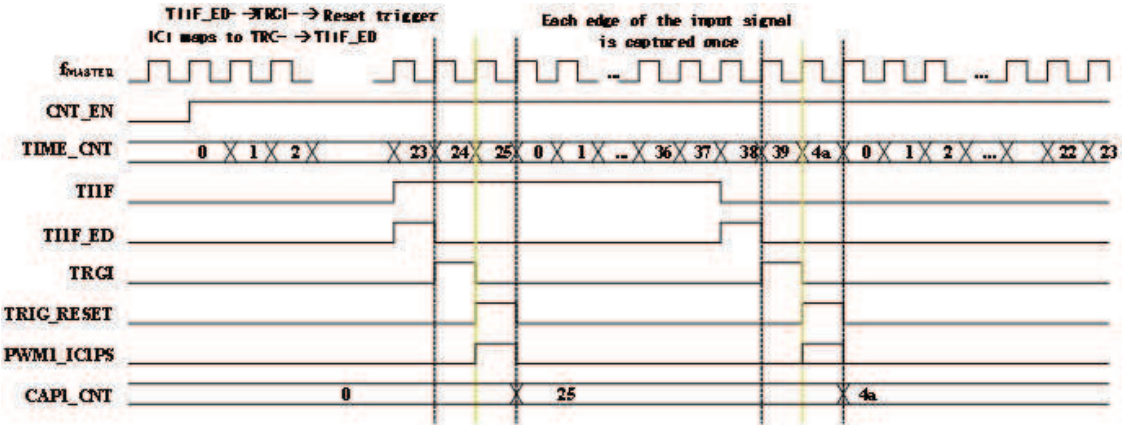


Figure 0.82 Input Capture Timing Sequence

8.2.4.3 Forced Output Mode

In output mode (CCxS=0 in the PWM1_CCMRx register), the output compare signal can be directly forced to a high or low state by software, independent of the comparison result between the output compare register and the counter.

The output comparison signal is forced at active by setting the corresponding OCxM=101 in the PWM1_CCMRx register. Thus OCxREF is forced at high (OCxREF is always active high), and the output of OCx is high or low depending on the CCxP polarity flag bit.

Set OCxM=100 in the PWM1_CCMRx register to force the OCxREF signal to low.

In this mode, the comparison between the PWM1_CCRx shadow register and the counter is still ongoing, the corresponding flags are modified, and the interrupts are still generated accordingly.

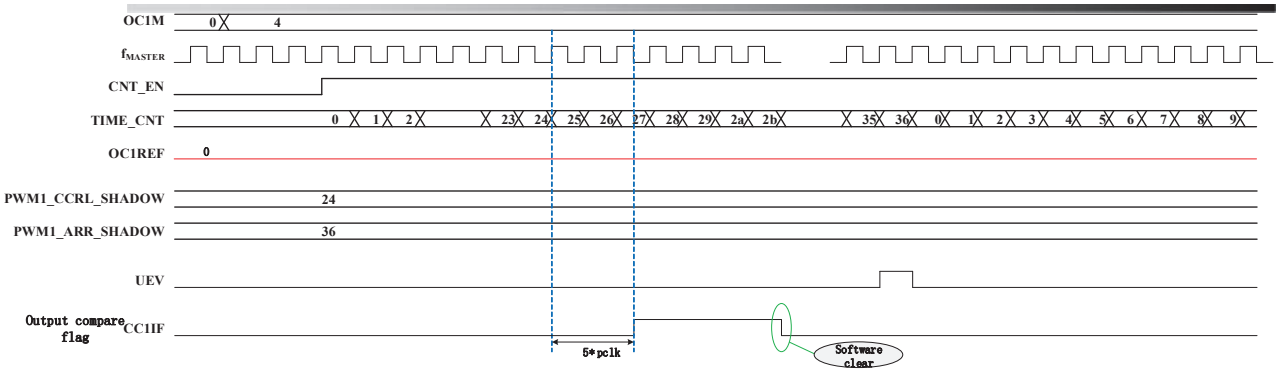


Figure 0.83 Forced output inactive level

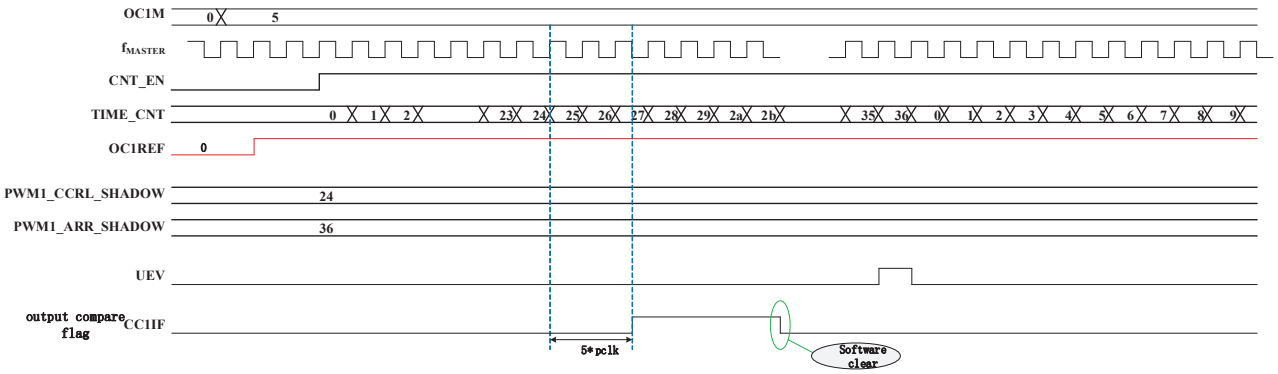


Figure 0.84 Forced output active level

For details, please refer to 16-bit advanced timer PWM0 Forced Output Mode.

8.2.4.4 Output Compare Mode

This mode is used to control an output waveform or to indicate that a given period of time has been reached.
When the counter has the same contents as the capture/compare register (CCRx), operations as follows:

- According to the different output compare modes, the corresponding OCx outputs signals,
  - Hold (OCxM=000)
  - Set to active level (OCxM=001)
  - Set to inactive level (OCxM=010)
  - Reverse (OCxM=011)
- Set the flag bit in the interrupt status register (CCxIF bit in the PWM1_SR1 register).
- If the corresponding interrupt enable bit (CCxIE bit in PWM1_IER register) is set, an interrupt is generated.

In output compare mode, the update event has no effect on the OCxREF and OCx outputs. The time accuracy is one cycle of the counter.

Configuration steps for output compare mode:

1. Selects the counter clock (internal, external, prescaler).
2. Write the corresponding data into the PWM1_ARR and PWM1_CCRx registers.
3. If an interrupt request is to be generated, set the CCxIE bit.
4. Select output mode steps:
    1) Reverse the output pin of OCxM when the counter matches CCRx, set OCxM=011
    2) Disable preload register by setting OCxPE=0
    3) Set CCxP=0 to select high level as active level
    4) Set CCxE=1 to enable output

Set the CEN bit of the PWM1_CR1 register to start the counter

The PWM1_CCRx register can be updated by software at any time to control the output waveform, provided that the pre-load register is not used (OCxPE='0', otherwise the shadow register of PWM1_CCRx can only be updated when the next update event occurs).

For details, please refer to 16-bit advanced timer PWM0 Output Compare Mode.8.1.4.6

8.2.4.5 PWM Mode

The pulse width modulation (PWM) mode generates a signal with a frequency determined by the PWM1_ARR register and a duty cycle determined by the PWM1_CCRx register.

Writing '110' (PWM mode 1) or '111' (PWM mode 2) to the OCxM bit in the PWM1_CCMRx register enables each OCx output channel to generate one PWM independently. The OCxPE bit of the PWM1_CCMRx register must be set to enable the corresponding preload register, or the ARPE bit of the PWM1_CR1 register can be set to enable the preload register for automatic reload (in up-counting mode or center-symmetric mode).

Note: Since the preload registers can be passed to the shadow registers only when an update event occurs, all registers must be initialized by setting the UG bit of the PWM1_EGR register before the counter starts counting.

The polarity of OCx can be set via software by the CCxP bit in the PWM1_CCERx register, which can be set to active high or active low. The output enable of OCx is controlled through CCxE in the PWM1_CCERx register. See the description of the PWM1_CCERx register for details.

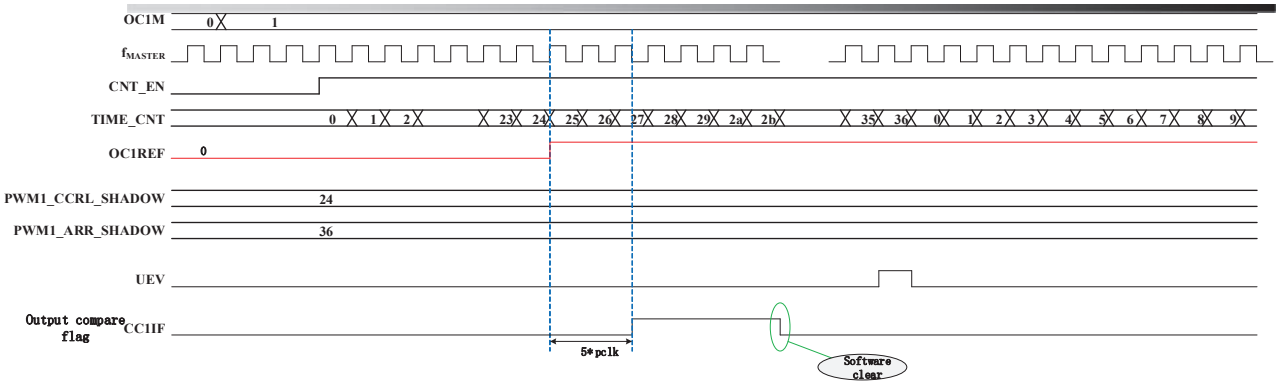


Figure 0.85 Output Compare (OCxM=0001)

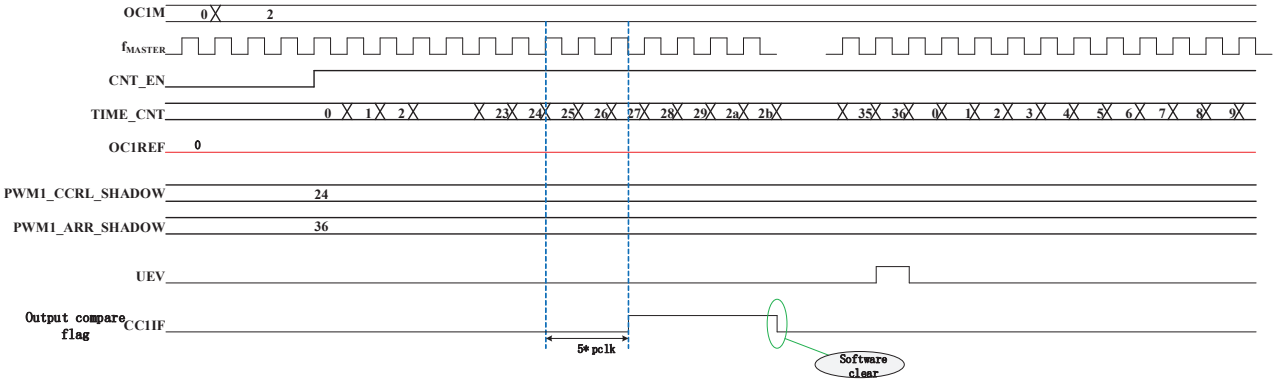


Figure 0.86 Output Compare (OCxM=0010)

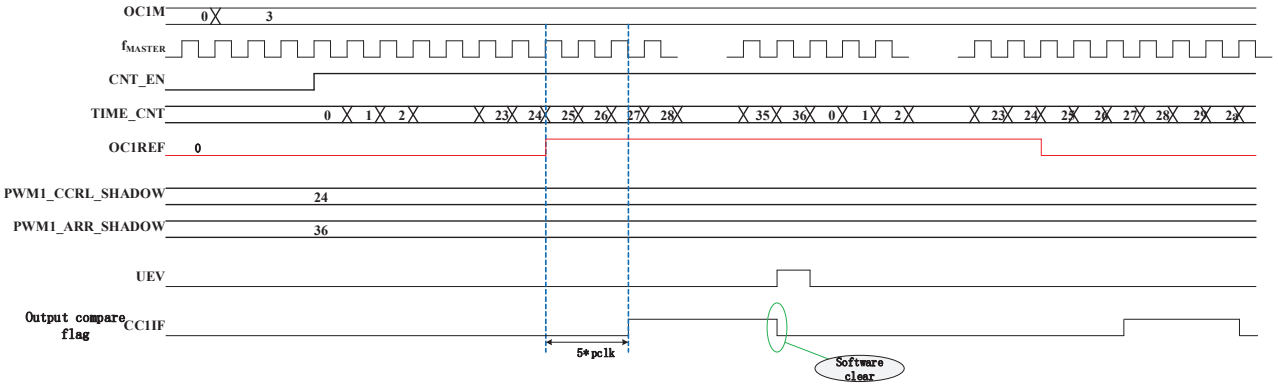


Figure 0.87 Output Compare (OCxM=0011)

For details, please refer to 16-bit Advanced Timer PWM0 Mode.

8.2.4.5.1 PWM Edge Aligned Mode

Up-counting configuration

In PWM mode 1, the PWM reference signal OCxREF is high when PWM1_CNT< PWM1_CCRx, otherwise it is low. If the comparison value in PWM1_CCRx is greater than the auto-reload value (PWM1_ARR), OCxREF remains '1'. If the compare value is 0, OCxREF remains '0'.
In PWM mode 2, the PWM reference signal OCxREF is low when PWM1_CNT< PWM1_CCRx, otherwise it is high. If the compare value in PWM1_CCRx is greater than the Auto-reload value (PWM1_ARR), OCxREF remains '0'. If the comparison value is 0, OCxREF remains '1'.

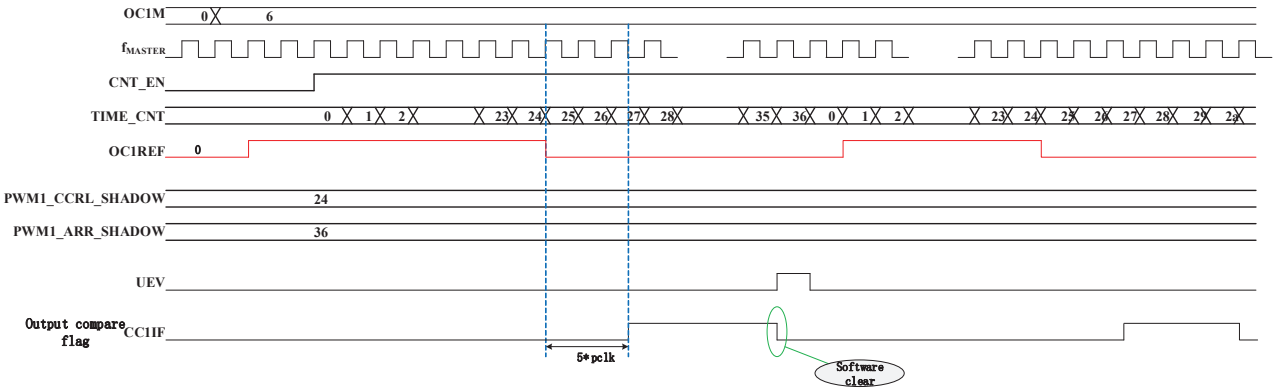


Figure 0.88 PWM Edge Aligned Mode 1 (Count Up Configuration)

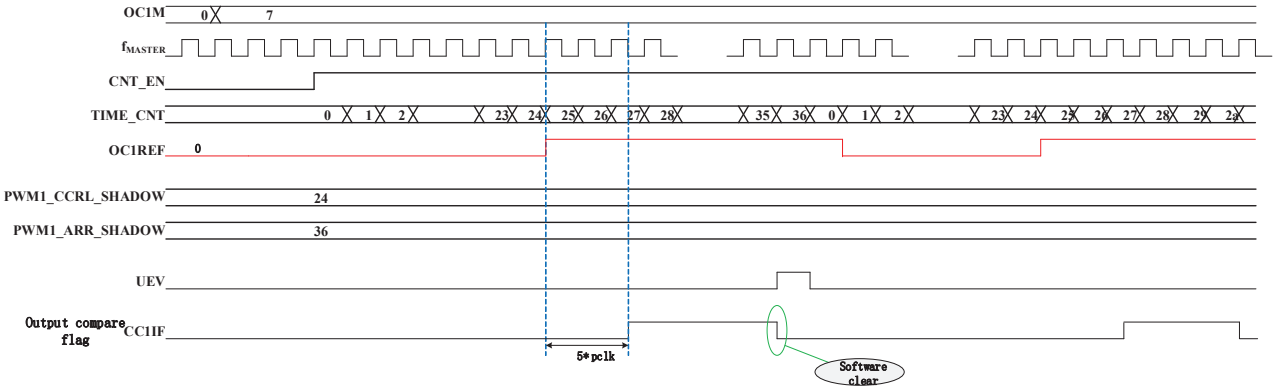


Figure 0.89 PWM Edge Aligned Mode 2 (Count Up Configuration)

8.2.4.5.2 One Pulse Mode

The one pulse mode (OPM) is a special case of the many modes described earlier. This mode allows the counter to respond to an stimulus and generate a pulse with controllable pulse width after a programmable delay.

The counter can be started by the clock/trigger controller and generate waveforms in output compare mode or PWM mode. Setting the OPM bit of the PWM1_CR1 register will select the one pulse mode, when the counter automatically stops at the next UEV.

A pulse can be generated only if the compare value is different from the initial value of counter. Before starting (when the timer is waiting to be triggered), it must be configured as follows.

- Up-counting mode: CNT < CCRx ≤ ARR.
- Down-counting mode: CNT > CCRx.

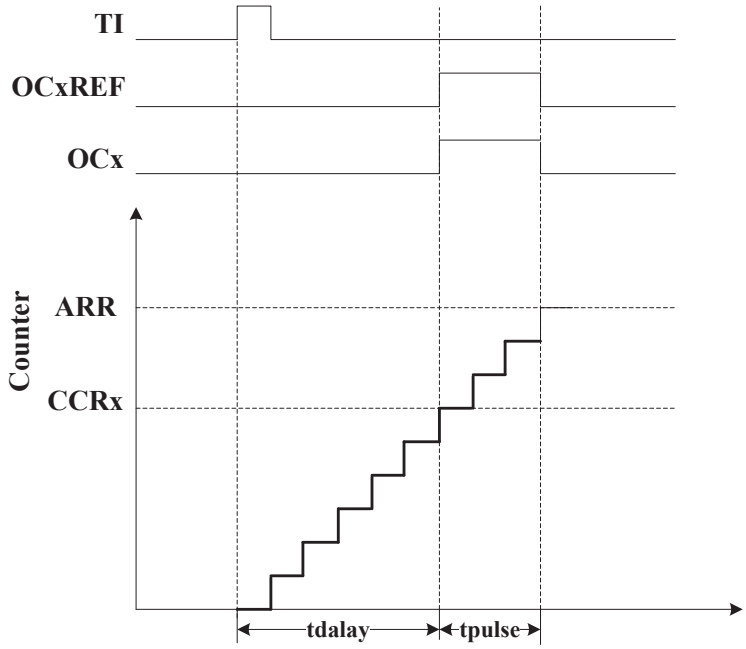


Figure 0.90 One Pulse Mode

Note: In one pulse mode, the configuration of the UG bit should be delayed for a period of time before the CEN bit can be configured, because the UG bit will pull down the CEN bit in hardware after a period of time.

8.2.4.6 Encoder Interface Mode

This mode is generally used for motor control. The encoder interface mode is selected by setting SMS=001 in the PWM1_SMCR register if the counter counts only on the edge of TI2, by setting SMS=010 if it counts only on the edge of TI1, or by setting SMS=011 if the counter counts on both the TI1 and TI2 edges.

Setting the CC1P and CC2P bits in the PWM1_CCER1 register allows you to select TI1 and TI2 polarities, you can also program the input filter if you would like.

The two inputs TI1 and TI2 are used as interfaces to the incremental encoder. The relationship between the count direction and the encoder signal is shown in the following table.

Effective edge	Signal level (TI1FP1 corresponds to TI2, TI2FP2 corresponds to TI1)	TI1FP1 signal		TI2FP2 signal	
		Rising	Falling	Rising	Falling
Count at TI1 only	High	Count down	Count up	No count	No count
	Low	Count up	Count down	No count	No count
Count at TI2 only	High	No count	No count	Count up	Count down
	Low	No count	No count	Count down	Count up
Count at TI1 and TI2	High	Count down	Count up	Count up	Count down
	Low	Count up	Count down	Count down	Count up

Table 0 .91 Relationship between count direction and encoder signal

Assume that the counter has been started (CEN=1 in the PWM1_CR1 register), the counter counts each time a valid jump is generated on TI1FP1 or TI2FP2=TI2. TI1FP1 and TI2FP2 are the signals of TI1 and TI2 after passing through the input filter and polarity control; if there is no filter and polarity shift, TI1FP1 = TI1, TI2FP2 = TI2. And the counting pulse and direction signal are generated according to the jump order of the two input signals. Also, the counter counts up or down while the hardware sets the DIR bit of the PWM1_CR1 register accordingly based on the jump order of the two input signals. Regardless of whether the counter counts on TI1, on TI2, or on both TI1 and TI2, the DIR bit is recalculated on the jump edge of either input (TI1 or TI2).

The encoder interface mode is basically similar to an external clock with direction selection. This means that the counter only counts continuously between 0 and the auto-reload value of the PWM1_ARR register (either 0 to ARR counting or ARR to 0 counting, depending on the direction). So PWM1_ARR must be configured before counting can start; again, the catcher, comparator, prescaler, trigger output characteristics still work normally.

8.2.5 Interrupt

PWM1 has 5 interrupt request sources:

1. Trigger interrupt
2. Input capture/output compare interrupt 3
3. Input capture/output compare interrupt 2
4. Input capture/output compare interrupt 1
5. Update event interrupt (e.g. counter overflow)

To use the interrupt feature, set the corresponding interrupt enable bits in the PWM1_IER register for each interrupt channel being used: i.e., the TIE, CCxIE, and UIE bits.

Each of the above interrupt sources can also be generated in software by setting the corresponding bit in the PWM1_EGR register.

8.3 PWM Registers

8.3.1 PWM0 Registers

8.3.1.1 Control Register 1(PWM0_CR1)

Address: 0xC3

Bit No.	7	6	5	4
Symbol	ARPE	CMS		DIR
Read/Write	Read/Write	Read/Write		Read/Write
Po Initial Value	0	0		0
Bit No.	3	2	1	0
Symbol	OPM	URS	UDIS	CEN
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	ARPE	Auto-preload enable bit. 0: PWM0_ARR has no buffer and can be directly written into. 1: PWM0_ARR is buffered by a preload buffer.
6~5	CMS	Selection center aligned mode. 00: edge aligned mode. Counter counts up or down according to DIR bit. 01: center aligned mode 1. counter counts up and down alternatively, The output compare interrupt flag bit of the channel configured as output (CCxS=00 in PWM0_CCRx register) is set to 1 only when the counter counts down. 10: Central aligned mode 2. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in PWM0_CCRx register) is set to 1 only when the counter counts up. 11: Central alignment mode 3. The counter counts up and down alternately. The output compare interrupt flag bit of the channel

		configured as output (CCxS=00 in PWM0_CCRx register) is set to 1 when the counter counts up and down. Note 1: When the counter is on (CEN=1), the transition from edge-aligned mode to center-aligned mode is not allowed. Note 2: In the central aligned mode, the encoder mode (SMS=001, 010, 011 in PWM0_SMCR register) must be disabled.
4	DIR	Directions. 0: counter counts up. 1: counter counts down. Note: This bit is read-only when the counter is configured in central aligned mode or encoder mode.
3	OPM	One pulse mode. 0: the counter does not stop when an update event occurs. 1: The counter stops when the next update event (clearing the CEN bit) occurs.
2	URS	Update the request source. 0: If UDIS allows the generation of update events, an update interrupt is generated by any of the following events. 1) Registers are updated (counter overflow/underflow) 2) Software setting UG bit 3) Updates generated by the clock/trigger controller 1: If UDIS allows the generation of update events, the update interrupt will be generated and UIF set to 1 only when the following events occur: the register is updated (counter overflow/underflow)
1	UDIS	Updates are prohibited. 0: Generates an update (UEV) event once the following events occur. 1) Counter overflow/underflow 2) Generate software update events 3)Hardware reset generated by the clock/trigger mode controller. The cached registers are loaded with their preloaded values. 1: No update event is generated and the shadow registers (ARR, PSC, CCRx) hold their values. The counters and prescaler are reinitialized if the UG bit is set or if the clock/trigger controller issues a hardware reset.
0	CEN	Counter enable. 0: disables the counter. 1: Enables the counter. Note: The external clock, gated mode and encoder mode will not work until the CEN bit is set in software. However, the trigger mode can automatically set the CEN bit by

		hardware.

8.3.1.2 Control register 2 (PWM0_CR2)

Address: 0xC4

Bit No.	7	6	5	4
Symbol	TI1S	MMS		
Read/Write	Read/Write	Read/Write		
Po Initial Value	0	0		
Bit No.	3	2	1	0
Symbol	-	COMS	-	CCPC
Read/Write	-	Read/Write	-	Read/Write
Po Initial Value	-	0	-	0

Bit No.	Symbol	Description
7	TI1S	TI1 selection. 0: CC1 input pin connected to TI1 (input of the digital filter). 1: CC1, CC2 and CC3 pins are connected to TI1 after XOR.
6~4	MMS	Master mode selection. This bit is used to select the synchronization information (TRGO) sent to the ADC or other slave timers in master mode. The possible combinations are as follows: 000: Reset. The UG bit of the PWM0_EGR register is used as the trigger output (TRGO). If the trigger input (clock/trigger controller configured for reset mode) generates a reset, there is a delay in the signal on TRGO relative to the actual reset. 001: Enable. The counter enable signal is used as a trigger output (TRGO), which is used to start multi-timers or ADCs in order to enable slave timer or ADC in a period of time. The counter enable signal is generated by the logical of the trigger input signal in the CEN control bit and gated mode. Unless Master/Slave mode is selected (see the description of the MSM bit in the PWM0_SMCR register), there is a delay on the TRGO when the counter enable signal is controlled by the trigger input. 010: Update. The update event is selected as trigger output. 011: Compare pulse (MATCH1). Once a capture or a successful comparison has occurred, the trigger output sends a positive pulse (TRGO) when the CC1IF flag is set to 1 (even if it is already high). 100: Compare. The OC1REF signal is used as the trigger output (TRGO). 101: Compare. The OC2REF signal is used as the trigger output

		(TRGO). 110: Compare. The OC3REF signal is used as the trigger output (TRGO). 111: Compare. The OC4REF signal is used as the trigger output (TRGO).
2	COMS	Update control selection for capture/compare control bit. 0: when the control bits for capture/compare are preloaded (CCPC=1), these control bits are updated only when COMG sets to 1. 1: when the capture/compare control bits are preloaded (CCPC=1), these control bits are updated only when a rising edge occurs at TRGI or COMG sets to 1. Note: This bit is only valid for channels that have complementary outputs.
0	CCPC	Capture/compare preload control bit. 0: CCxE, CCxNE, CCxP, CCxNP bits (PWM0_CCERx register),OCxM bits (PWM0_CCMRx register) are not preloaded. 1: The CCxE, CCxNE, CCxP, CCxNP and OCxM bits are preloaded; when this bit is set, they are updated only after the COMG bit (PWM0_EGR register) is set. Note: This bit only works for channels with complementary outputs.

8.3.1.3 Slave Mode Control Register (PWM0_SMCR)

Address: 0xC5

Bit No.	7	6	5	4
Symbol	MSM	TS		
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	0	0	0	
Bit No.	3	2	1	0
Symbol	-	SMS		
Read/Write	-	Read/Write		
Po Initial Value	-	0		

Bit No.	Symbol	Description
7	MSM	Master/Slave Mode. 0: no effect. 1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between Timer 1 and its slave timer (via TRGO).
6~4	TS	Trigger selection, used to select the trigger input for the

		synchronous counter. 000: Reserved 001: Reserved 010: Internal trigger ITR connected to PWM1 TRGO 011: Reservation 100: Edge detector for TI1 (TI1F_ED) 101: Timer input after filtering (TI1FP1) 110: Timer input after filtering (TI2FP2) 111: External trigger input (ETRF) Note: These bits can only be changed when they are not used (e.g. SMS=000) to avoid false edge detection when they are changed.
2~0	SMS	Clock/Trigger/Slave Mode Selection. When an external signal is selected, the active edge of the trigger signal (TRGI) is related to the polarity of the selected external input (see Description of the Input Control Register and Control Register) 000: Clock/trigger controller disabled. If CEN = 1, the prescaler is driven directly by the internal clock. 001: Encoder mode 1. Depending on the level of TI1FP1, the counter counts up/down on the edge of TI2FP2. 010: Encoder mode 2. Depending on the level of TI2FP2, the counter counts up/down on the edge of TI1FP1. 011: Encoder mode 3. Depending on the level of the other input, the counter counts up/down on the edges of TI1FP1 and TI2FP2. 100: Reset mode. The counter is reinitialized on the rising edge of the selected trigger input (TRGI) and a signal is generated to update the register. 101: Gate control mode. When the trigger input (TRGI) is high, the counter is clocked on. Once the trigger input becomes low, the counter stops (but does not reset). The counter is started and stopped in a controlled manner. 110: Trigger mode. The counter is started (but not reset) on the rising edge of the trigger input TRGI, and only the start of the counter is controlled. 111: External clock mode 1. The rising edge of the selected trigger input (TRGI) drives the counter. Note: Do not use gate control mode if TI1F_ED is selected as the trigger input (TS=100). This is because TI1F_ED only outputs a pulse each time TI1F changes, however the gated mode is to check the level of the trigger input.

8.3.1.4 External Trigger Register (PWM0_ETR)

Address: 0xC6

Bit No.	7	6	5	4
Symbol	ETP	ECE	ETPS	
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	0	0	0	
Bit No.	3	2	1	0
Symbol	ETF			
Read/Write	Read/Write			
Po Initial Value	0			

Bit No.	Symbol	Description
7	ETP	External trigger polarity. This bit determines whether the ETR or the inverse of the ETR is used to trigger the operation. 0: ETR is not inverted, i.e. high or rising edge active. 1: ETR is inverted, i.e. low or falling edge is active.
6	ECE	External Clock Enable. This bit is used to enable external clock mode 2. 0: disabled. 1: Enable external mode 2, the counter is clocked to the active edge of ETRF. Note 1: The effect of ECE set 1 is the same as the option to connect TRGI to external clock mode 1 of ETRF (SMS=111 and TS=111 in PWM0_SMCR register). Note 2: External mode 2 can be used simultaneously with the following modes: trigger standard mode, trigger reset mode, trigger gated mode. However, TRGI must not be connected to ETRF at this time (in PWM0_SMCR register, TS cannot be 111)
5~4	ETPS	External trigger prescaler. The maximum frequency of the external trigger signal ETRP cannot exceed $f_{MASTER}/4$. A prescaler can be used to reduce the frequency of ETRP, which is very useful when the frequency of ETRP is high. 00: Prescaler off. 01: Frequency of ETRP/2. 10: Frequency of ETRP/4. 11: Frequency of ETRP/8.
3~0	ETF	External trigger filter selection.

		The sampling frequency of the ETRP and the length of the digital filter are defined. The digital filter consists of an event counter that records N events and then generates a jump for the output. 0000: No filter, sampled with f_{MASTER} 0001: Sampling frequency $f_{SAMPLING}=f_{MASTER}$, N=2 0010: Sampling frequency $f_{SAMPLING}=f_{MASTER}$, N=4 0011: Sampling frequency $f_{SAMPLING}=f_{MASTER}$, N=8 0100: Sampling frequency $f_{SAMPLING}=f_{MASTER}/2$, N=6 0101: Sampling frequency $f_{SAMPLING}=f_{MASTER}/2$, N=8 0110: Sampling frequency $f_{SAMPLING}=f_{MASTER}/4$, N=6 0111: Sampling frequency $f_{SAMPLING}=f_{MASTER}/4$, N=8 1000: Sampling frequency $f_{SAMPLING}=f_{MASTER}/8$, N=6 1001: Sampling frequency $f_{SAMPLING}=f_{MASTER}/8$, N=8 1010: Sampling frequency $f_{SAMPLING}=f_{MASTER}/16$, N=5 1011: Sampling frequency $f_{SAMPLING}=f_{MASTER}/16$, N=6 1100: Sampling frequency $f_{SAMPLING}=f_{MASTER}/16$, N=8 1101: Sampling frequency $f_{SAMPLING}=f_{MASTER}/32$, N=5 1110: Sampling frequency $f_{SAMPLING}=f_{MASTER}/32$, N=6 1111: Sampling frequency $f_{SAMPLING}=f_{MASTER}/32$, N=8

8.3.1.5 Interrupt Enable Register (PWM0_IER)

Address: 0xC7

Bit No.	7	6	5	4
Symbol	BIE	TIE	COMIE	CC4IE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	CC3IE	CC2IE	CC1IE	UIE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	BIE	Break interrupt enable. 0: disabled 1: enabled
6	TIE	Trigger Interrupt Enable.

		0: disabled 1: enabled
5	COMIE	COM interrupt enable. 0: disabled 1: enabled
4	CC4IE	Capture/Compare 4 interrupt enable. 0: disabled 1: enabled
3	CC3IE	Capture/Compare 3 interrupt enable.0: disabled 1: enabled
2	CC2IE	Capture/Compare 2 interrupt enable.0: disabled 1: enabled
1	CC1IE	Capture/Compare 1 interrupt enable.0: disabled 1: enabled
0	UIE	Update Interrupt enable. 0: disabled 1: enabled

8.3.1.6 Status Register 1 (PWM0_SR1)

Address: 0xC8

Bit No.	7	6	5	4
Symbol	BIF	TIF	COMIF	CC4IF
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	CC3IF	CC2IF	CC1IF	UIF
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	BIF	Break Interrupt flag. Once the break is valid, the bit is set to 1 by hardware. If the break input is invalid, the bit can be written 0 and cleared to 0 by software. 0: No break events. 1: Active level is detected on the break input.

6	TIF	Trigger interrupt flag. When a trigger event occurs (a active edge is detected at the TRGI input when the slave mode controller is in a mode other than gated mode. Or any edge in gated mode), hardware sets it to 1. The bit can be written 0 and cleared to 0 by software. 0: no trigger events occur. 1: Trigger an interrupt to wait for response.
5	COMIF	COM interrupt flag. Once a COM event is generated (when the capture/compare control bits: CCxE, CCxNE, OCxM have been updated) this bit is set to 1 by hardware. it is written to 0 or cleared to 0 by software. 0: no COM events generated. 1: COM interrupt waiting
4	CC4IF	Capture/compare 4 interrupt flag. Refer to CC1IF description.
3	CC3IF	Capture/compare 3 interrupt flag. Refer to CC1IF description.
2	CC2IF	Capture/compare 2 interrupt flag. Refer to CC1IF description.
1	CC1IF	Capture/compare 1 interrupt flag. If channel CC1 is configured as output mode. This bit is set to 1 by hardware when the counter value matches the comparison value, except in center aligned mode (refer to the CMS bit of the PWM0_CR1 register). it is written to 0 or cleared to 0 by software. 0: no match occurs. 1: The value of PWM0_CNT matches the value of PWM0_CCR1. Note: In center aligned mode, when the counter is 0, it counts up, and when the counter is ARR, it counts down (it counts up from 0 to ARR-1, and then from ARR counts down to 1). Therefore, for all SMS values, both values are not flagged. However, if CCR1>ARR, CC1IF is set to 1 when CNT reaches ARR. If channel CC1 is configured as input mode. This bit is set to 1 by hardware when a capture event occurs, and it is written to 0/cleared to 0 by software or by reading PWM0_CCR1 to clear to 0. 0: no input capture generates. 1: The counter value has been captured (copied) to PWM0_CCR1 (an edge with the same polarity as the selected one is detected on IC1).
0	UIF	Update interrupt flag. This bit is set to 1 by hardware when an update event is generated. It is written to 0 or cleared to 0 by

		software. 0: no update events generated. 1: Update event wait accordingly. This bit is set to 1 by hardware when the register is updated. -- if UDIS=0 in the PWM0_CR1 register, when the register overflows or underflows. -- if UDIS=0 and URS=0 of PWM0_CR1 register when setting the UG bit of PWM0_ETR register software to reinitialize the counter CNT. -- If UDIS=0 and URS=0 of PWM0_CR1 register, when the counter CNT is reinitialized by a trigger event (refer to slave mode controller PWM0_SMCR).

8.3.1.7 Status Register 2 (PWM0_SR2)

Address: 0xC9

Bit No.	7	6	5	4
Symbol	-			CC4OF
Read/Write	-			Read/Write
Po Initial Value	-			0
Bit No.	3	2	1	0
Symbol	CC3OF	CC2OF	CC1OF	-
Read/Write	Read/Write	Read/Write	Read/Write	-
Po Initial Value	0	0	0	-

Bit No.	Symbol	Description
4	CC4OF	Capture/compare 4 repetition capture flag See CC1OF description.
3	CC3OF	Capture/compare 3 repetition capture flag See CC1OF description.
2	CC2OF	Capture/compare 2 repetition capture flag See CC1OF description.
1	CC1OF	Capture/compare 1 repetition capture flag This flag can be set to 1 by hardware only when the corresponding channel is configured for input capture. Writing 0 clears the bit. 0: No repetition capture is generated. 1: The value of the counter is captured to the PWM0_CCR1 register

		when the status of CC1IF is already 1.

8.3.1.8 Event Generation Register (PWM0_EGR)

Address: 0xCA

Bit No.	7	6	5	4
Symbol	BG	TG	COMG	CC4G
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	CC3G	CC2G	CC1G	UG
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	BG	Generate a break event. This bit is set to 1 by software to generate a break event, which is automatically cleared to 0 by hardware. 0: no behavior. 1: Generate a break event. At this time, MOE=0 and BIF=1. If the corresponding interrupt is on (BIE=1), the corresponding interrupt will be generated.
6	TG	Generate a trigger event. This bit is set to 1 by software to generate a trigger event that is automatically cleared to 0 by hardware. 0: no behavior. 1: TIF=1 of PWM0_SR register, if the corresponding interrupt is turned on (TIE=1), the corresponding interrupt is generated.
5	COMG	Capture/compare events and generate control updates. This bit is set to 1 by software and cleared to 0 automatically by hardware. 0: no behavior. 1: When CCPC=1, it is allowed to update CCxE, CCxNE, CCxP, CCxNP, OCxM bits. Note: This bit is only valid for channels that have complementary outputs.
4	CC4G	Generate capture/compare 4 events. Refer to CC1G description.
3	CC3G	Generate capture/compare 3 events.

		Refer to CC1G description.
2	CC2G	Generate capture/compare 2 events. Refer to CC1G description.。
1	CC1G	Generate capture/compare 1 events. This bit is set to 1 by software to generate a capture/compare event, which is automatically cleared to 0 by hardware. 0: no behavior. 1: Generate a capture/compare event on channel CC1. If channel CC1 is configured as output. Set CC1IF=1 to generate the corresponding interrupt if the corresponding interrupt is turned on. If channel CC1 is configured as input. The current counter value is captured to the PWM0_CCR1 register, set CC1IF=1, and if the corresponding interrupt is turned on, the corresponding interrupt is generated. If CC1IF is already 1, then set CC1OF=1.
0	UG	Generates an update event. This bit is set to 1 by software and automatically cleared to 0 by hardware. 0: no behavior. 1: Reinitialize the counter and generate an update event. Note that the counter of the prescaler is also cleared to 0 (but the prescaler factor is unchanged). The counter is cleared to 0 if in center aligned mode or if DIR=0 (counting up); if DIR=1 (counting down) the counter takes the value of PWM0_ARR.

8.3.1.9 Capture/Compare Mode Register 1 (PWM0_CCMR1)

The channel can be used for input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CC1S bit. The other bits of this register have different functions in input and output mode. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode. Therefore it must be noted that the same bit has a different function in output mode and in input mode.

Address: 0xCB

- Input Capture Mode

Bit No.	7	6	5	4
Symbol	IC1F			
Read/Write	Read/Write			
Po Initial Value	0			

Bit No.	3	2	1	0
Symbol	IC1PSC		CC1S	
Read/Write	Read/Write		Read/Write	
Po Initial Value	0		0	

Bit No.	Symbol	Description
7~4	IC1F	Input capture 1 filter. These bits define the sampling frequency of the TI1 input and the size of the digital filter. The digital filter consists of an event counter, and the output jumps are considered valid only after N events have occurred. 0000: No filter, $f_{SAMPLING}=f_{MASTER}$ 0001: $f_{SAMPLING}=f_{MASTER}$, N=2 0010: $f_{SAMPLING}=f_{MASTER}$, N=4 0011: $f_{SAMPLING}=f_{MASTER}$, N=8 0100: $f_{SAMPLING}=f_{MASTER}/2$, N=6 0101: $f_{SAMPLING}=f_{MASTER}/2$, N=8 0110: $f_{SAMPLING}=f_{MASTER}/4$, N=6 0111: $f_{SAMPLING}=f_{MASTER}/4$, N=8 1000: $f_{SAMPLING}=f_{MASTER}/8$, N=6 1001: $f_{SAMPLING}=f_{MASTER}/8$, N=8 1010: $f_{SAMPLING}=f_{MASTER}/16$, N=5 1011: $f_{SAMPLING}=f_{MASTER}/16$, N=6 1100: $f_{SAMPLING}=f_{MASTER}/16$, N=8 1101: $f_{SAMPLING}=f_{MASTER}/32$, N=5 1110: $f_{SAMPLING}=f_{MASTER}/32$, N=6 1111: $f_{SAMPLING}=f_{MASTER}/32$, N=8 Note: Even for channels with complementary outputs, this bit field is non-preloaded and does not take into account the value of CCPC (PWM0_CR2 register).
3~2	IC1PSC	Input/capture 1 prescaler. These 2 bits define the prescaler factor of the CC1 input (IC1). Once CC1E=0 (in PWM0_CCER register), the prescaler is reset. 00: without prescaler, one capture is triggered for each edge detected on the capture input. 01: capture triggered every 2 events. 10: capture triggered every 4 events. 11: Capture is triggered every 8 events.
1~0	CC1S	Capture/Compare 1 Selection. These 2 bits define the direction of the channel (input/output), and the selection of the input pin.

		00: CC1 channel is configured as an output. 01: CC1 channel is configured as input and IC1 is mapped on TI1FP1. 10: CC1 channel is configured as input and IC1 is mapped on TI2FP1. 11: CC1 channel is configured as input and IC1 is mapped on TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM0_SMCR register). Note: CC1S can be written only when the channel is off (PWM0_CCER1 register CC1E=0, CC1NE=0 and has been updated).

- **Output Compare Mode**

Bit No.	7	6	5	4
Symbol	OC1CE	OC1M		
Read/Write	Read/Write	Read/Write		
Po Initial Value	0	0		
Bit No.	3	2	1	0
Symbol	OC1PE	OC1FE	CC1S	
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	0	0	0	

Bit No.	Symbol	Description
7	OC1CE	Output compare 1 clear enable. This bit is used to enable the use of an external event on the ETR pin to clear the output signal of channel 1 (OC1REF). Refer to *Clear the OCREF signal when an external event occurs.* 0: OC1REF is not affected by the ETRF input (from the ETR pin). 1: OC1REF=0 once ETRF input is detected high.
6~4	OC1M	Output compare 1 mode. These 3 bits define the behavior of the output reference signal OC1REF which determines the value of OC1. OC1REF is active high, while the active level of OC1 depends on the CC1P bit. 000: Frozen. The comparison between the output compare register PWM0_CCR1 and the counter PWM0_CNT does not work for OC1REF. 001: Set the output of channel 1 to active level when matching. Force OC1REF to high when the value of counter PWM0_CNT is the same as capture/compare register 1 PWM0_CCR1).

		010: Set the output of channel 1 to inactive level when matching. Force OC1REF to low when the value of counter PWM0_CNT is the same as capture/compare register 1 (PWM0_CCR1). 011: reverse. Reverse the level of OC1REF when PWM0_CCR1= PWM0_CNT. 100: Forced to inactive level. Forces OC1REF to low. 101: Force to active level. Force OC1REF to high. 110: PWM mode 1- In count up, channel 1 is active once PWM0_CNT< PWM0_CCR1, otherwise it is inactive; in count down, channel 1 is inactive once PWM0_CNT> PWM0_CCR1 (OC1REF=0), otherwise it is active (OC1REF=1). 111: PWM mode 2- In count up, channel 1 is inactive once PWM0_CNT< PWM0_CCR1, otherwise it is active; in count down, channel 1 is active once PWM0_CNT> PWM0_CCR1, otherwise it is inactive. Note 1: Once the LOCK level is set to 3 (LOCK bit in PWM0_BKR register) and CC1S=00 (the channel is configured as an output) this bit cannot be modified. Note 2: In PWM mode 1 or PWM mode 2, the OC1REF level changes only when the comparison result is changed or when switching from froze mode to PWM mode in the output compare mode. (Reference *PWM mode*) Note 3: On channels with complementary outputs, these bits are preloaded. If the PWM0_CR2 register has CCPC=1, the OCM bit takes a new value from the preload bit only when a COM event occurs.
3	OC1PE	Output Compare 1 Preload Enable. 0: Preload function of PWM0_CCR1 register is disabled, PWM0_CCR1 register can be written at any time, and the newly written value takes effect immediately. 1: Enable the preload function of PWM0_CCR1 register, the read/write operation only operates on the preload register, and the preload value of PWM0_CCR1 is loaded into the current register when the update event comes. Note 1: Once the LOCK level is set to 3 (LOCK bit in PWM0_BKR register) and CC1S=00 (the channel is configured as an output) this bit cannot be modified. Note 2: In order to operate correctly, the preload function must be enabled in PWM mode. However, in onepulse mode (PWM0_CR1 register with OPM=1), it is not necessary.
2	OC1FE	Output Compare 1 Fast Enable. This bit is used to speed up the response of the CC output to a trigger

		input event. 0: Depending on the value of the counter and CCR1, CC1 operates normally, even if the trigger is on. The minimum delay to activate the CC1 output is 5 clock cycles when the input of the trigger has active edge. 1: The active edge of the input to the trigger acts as if a comparison match has occurred. Therefore, OC is set to the comparison level independent of the comparison result. The delay between the active edge of the sampling trigger and the CC1 output is reduced to 3 clock cycles. OC1FE only works when the channel is configured to PWM1 or PWM2 mode.
1~0	CC1S	CC1S[1:0]: Capture/Compare 1 Selection. These 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: CC1 channel is configured as output. 01: CC1 channel is configured as input and IC1 is mapped on TI1FP1. 10: CC1 channel is configured as input and IC1 is mapped on TI2FP1. 11: CC1 channel is configured as input and IC1 is mapped on TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM0_SMCR register). Note: CC1S can be written only when the channel is off (PWM0_CCER1 register CC1E=0, CC1NE=0 and has been updated).

8.3.1.10 Capture/Compare Mode Register 2 (PWM0_CCMR2)

Address: 0xCC

- **Input Capture Mode**

Bit No.	7	6	5	4
Symbol	IC2F			
Read/Write	Read/Write			
Po Initial Value	0			
Bit No.	3	2	1	0
Symbol	IC2PSC		CC2S	
Read/Write	Read/Write		Read/Write	

Po Initial Value	0	0

Bit No.	Symbol	Description
7~4	IC2F	Input capture 2 filter.
3~2	IC2PSC	Input capture 2 prescaler.
1~0	CC2S	Input capture 2 selection. These 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: CC2 channel is configured as output. 01: CC2 channel is configured as input and IC2 is mapped on TI2FP2. 10: CC2 channel is configured as input and IC2 is mapped on TI1FP2. 11: CC2 channel is configured as input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM0_SMCR register). Note: CC2S can be written only when the channel is off (CC2E=0 and CC2NE=0 of PWM0_CCER1 register and has been updated).

- Output Compare Mode

Bit No.	7	6	5	4
Symbol	OC2CE	OC2M		
Read/Write	Read/Write	Read/Write		
Po Initial Value	0	0		
Bit No.	3	2	1	0
Symbol	OC2PE	OC2FE	CC2S	
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	0	0	0	

Bit No.	Symbol	Description
7	OC2CE	Output Compare 2 Clear Enable. This bit is used to enable the use of an external event on the ETR pin to clear the output signal of channel 2 (OC2REF), with reference to *Clear the OCREF signal when an external event occurs* 0: OC2REF is not affected by the ETRF input (from the ETR pin). 1: OC2REF=0 once ETRF input is detected high.

6~4	OC2M	Output compare 2 mode.
3	OC2PE	Output Compare 2 Preload Enable.
2	OC2FE	Output compare 2 fast enable.
1~0	CC2S	The bit defines the direction of the channel (input/output) and the selection of the input pin. Capture/compare 2 selection. 00: CC2 is configured as output. 01: CC2 is configured as input, IC2 is mapped on TI2FP2. 10: CC2 is configured as input, IC2 is mapped on TI1FP2. 11: CC2 is configured as input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM0_SMCR register). Note: CC2S can be written only when the channel is off (CC2E=0 and CC2NE=0 of PWM0_CCER1 register and has been updated).

8.3.1.11 Capture/Compare Mode Register 3 (PWM0_CCMR3)

Address: 0xCD

- Input Capture Mode

Bit No.	7	6	5	4
Symbol	IC3F			
Read/Write	Read/Write			
Po Initial Value	0			
Bit No.	**3**	**2**	**1**	**0**
Symbol	IC3PSC		CC3S	
Read/Write	Read/Write		Read/Write	
Po Initial Value	0		0	

Bit No.	Symbol	Description
7~4	IC3F	Input capture 3 filter.
3~2	IC3PSC	Input/capture 3 prescaler.
1~0	CC3S	Input/capture 3 selection. These 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: CC3 channel is configured as output. 01: CC3 channel is configured as input and IC3 is mapped on TI3FP3. 10: CC3 channel is configured as input and IC3 is mapped on

		TI4FP3. 11: CC3 channel is configured as input and IC3 is mapped on TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM0_SMCR register). Note: CC3S can be written only when the channel is off (PWM0_CCER2 register CC3E=0, CC3NE=0 and has been updated).

- Output Capture Mode

Bit No.	7	6	5	4
Symbol	OC3CE	OC3M		
Read/Write	Read/Write	Read/Write		
Po Initial Value	0	0		
Bit No.	3	2	1	0
Symbol	OC3PE	OC3FE	CC3S	
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	0	0	0	

Bit No.	Symbol	Description
7	OC3CE	Output compare 3 clear enable. This bit is used to enable the use of an external event on the ETR pin to clear the output signal of channel 3 (OC3REF), with reference to *Clear the OCREF signal when an external event occurs.* 0: OC3REF is not affected by the ETRF input (from the ETR pin). 1: OC3REF=0 once ETRF input is detected high.
6~4	OC3M	Output compare 3 mode.
3	OC3PE	Output compare 3 preload enable.
2	OC3FE	Output compare 3 fast enable.
1~0	CC3S	Capture/compare 3 selection. This bit defines the direction of the channel (input/output), and the selection of the input pin. 00: CC3 channel is configured as output. 01: CC3 channel is configured as input and IC3 is mapped on TI3FP3. 10: CC3 channel is configured as input and IC3 is mapped on TI4FP3. 11: CC3 channel is configured as input and IC3 is mapped on TRC. This mode works only when the internal trigger input is selected

		(selected by the TS bit of the PWM0_SMCR register). Note: CC3S can be written only when the channel is off (PWM0_CCER2 register CC3E=0, CC3NE=0 and has been updated).

8.3.1.12 Capture/Compare Mode Register 4 (PWM0_CCMR4)

Address: 0xCE

- Input Capture Mode

Bit No.	7	6	5	4
Symbol	IC4F			
Read/Write	Read/Write			
Po Initial Value	0			
Bit No.	3	2	1	0
Symbol	IC4PSC		CC4S	
Read/Write	Read/Write		Read/Write	
Po Initial Value	0		0	

Bit No.	Symbol	Description
7~4	IC4F	Input capture 4 filter.
3~2	IC4PSC	Input/capture 4 prescaler.
1~0	CC4S	Capture/Compare 4 selection. These 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: CC4 channel is configured as output. 01: CC4 channel is configured as input and IC4 is mapped on the TI4FP4. 10: CC4 channel is configured as input and IC4 is mapped on TI3FP4. 11: CC4 channel is configured as input and IC4 is mapped on TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM0_SMCR register). Note: CC4S can be written only when the channel is off (CC4E=0 of PWM0_CCER2 register).

- Output Compare Mode

Bit No.	7	6	5	4

Symbol	OC4CE	OC4M		
Read/Write	Read/Write	Read/Write		
Po Initial Value	0	0		
Bit No.	3	2	1	0
Symbol	OC4PE	OC4FE	CC4S	
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	0	0	0	

Bit No.	Symbol	Description
7	OC4CE	Output compare 4 clear enable. This bit is used to enable the use of an external event on the ETR pin to clear the output signal of channel 4 (OC4REF), with reference to *Clear the OCREF signal when an external event occurs.* 0: OC4REF is not affected by the ETRF input (from the ETR pin). 1: OC4REF = 0 once ETRF input is detected high.
6~4	OC4M	Output compare 4 mode.
3	OC4PE	Output compare 4 preload enable.
2	OC4FE	Output compare 4 fast enable.
1~0	CC4S	Capture/Compare 4 selection. This bit defines the direction of the channel (input/output), and the selection of the input pin. 00: CC4 channel is configured as output. 01: CC4 channel is configured as input and IC4 is mapped on the TI4FP4. 10: CC4 channel is configured as input and IC4 is mapped on TI3FP4. 11: CC4 channel is configured as input and IC4 is mapped on TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM0_SMCR register). Note: CC4S can be written only when the channel is off (CC4E=0 of PWM0_CCER2 register).

8.3.1.13 Capture/Compare Enable Register 1 (PWM0_CCER1)

Address: 0xCF

Bit No.	7	6	5	4
Symbol	CC2NP	CC2NE	CC2P	CC2E
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write

Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	CC1NP	CC1NE	CC1P	CC1E
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	CC2NP	Input capture/compare 2 complementary output polarity. Refer to the description of CC1NP.
6	CC2NE	Input Capture/Compare 2 complementary output enable. Refer to the description of CC1NE.
5	CC2P	Input capture/compare 2 output polarity. Refer to the description of CC1P.
4	CC2E	Input Capture/Compare 2 output enable. Refer to the description of CC1E.
3	CC1NP	Input capture/compare 1 complementary output polarity. 0: OC1N is active high. 1: OC1N is active low. Note 1: This bit cannot be modified once the LOCK level (LOCK bit in PWM0_BKR register) is set to 3 or 2 and CC1S=00 (channel is configured as output). Note 2: For channels with complementary outputs, this bit is preloaded. If CCPC=1 (PWM0_CR2 register), the CC1NP bit takes a new value from the preload bit only when a COM event occurs.
2	CC1NE	Input Capture/Compare 1 complementary output enable. 0: Off - OC1N disables output, so the output level of OC1N depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1E bits. 1: ON- OC1N signal is output to the corresponding output pin and its output level depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1E bits. Note: For channels with complementary outputs, this bit is preloaded. If CCPC=1 (PWM0_CR2 register), the CC1NE bit takes a new value from the preload bit only when a COM event occurs.
1	CC1P	Input Capture/Compare 1 Output polarity. The CC1 channel is configured as output. 0: OC1 active high. 1: OC1 is active low. CC1 channel is configured as trigger (Refer to Figure 0.41 Inputs of PWM0 Channel 1). 0: The trigger occurs at a high level or rising edge of TI1F.

		1: The trigger occurs at the low level or falling edge of TI1F. The CC1 channel is configured as input (Refer to Figure 8.41) 0: Capture occurs on the high level or rising edge of TI1F. 1: Capture occurs on the low level or falling edge of TI1F. Note 1: Once the LOCK level (LOCK bit in PWM0_BKR register) is set to 3 or 2 and CC1S=00 (channel is configured as output), this bit cannot be modified. Note 2: For channels with complementary outputs, this bit is preloaded. If CCPC=1 (PWM0_CR2 register), the CC1P bit takes a new value from the preload bit only when a COM event occurs.
0	CC1E	Input Capture/Compare 1 output enable. CC1 channel is configured as output. 0: Off - OC1 disables output, so the output level of OC1 depends on the values of the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits. 1: ON- OC1 signal is output to the corresponding output pin and its output level depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits. CC1 channel is configured as input. This bit determines whether the counter value can be captured into the PWM0_CCR1 register. 0: Capture disabled. 1: Capture enabled. Note: For channels with complementary outputs, this bit is preloaded. If CCPC=1 (PWM0_CR2 register), the CC1E bit takes a new value from the preload bit only when a COM event occurs.

Control bit					Control bit	
MOE bit	**OSSI bit**	**OSSR bit**	***CCxE bit***	***CCxNE bit***	**OC*x output status***	**OC*x*N output status**
1	X	0	0	0	Output disabled (disconnected from timer)	Output disabled (disconnected from timer)
		0	0	1	Output disabled (disconnected from timer)	OCxREF + polarity. OCxN= OCxREF xor CCxNP
		0	1	0	OCxREF + polarity. OCx= OCxREF xor CCxP	Output disabled (disconnected from timer)
		0	1	1	OC*x*REF + polarity	OC*x*REF inverted +

					+ deadtime	polarity + deadtime
		1	0	0	Output disabled (disconnected from timer)	Output disabled (disconnected from timer)
		1	0	1	OFF (output enabled and inactive level) OC*x*=CCxP	OCxREF + polarity. OCxN= OCxREF xor CCxNP
		1	1	0	OC*x*REF + polarity, OC*x*= OC*x*REF xor CC*x*P	OFF (output enabled and inactive level) OC*x*N=CC*x*NP
		1	1	1	OC*x*REF + polarity + deadtime	OC*x*REF inverted + polarity + deadtime
0	0	X	X	X	Output disabled (disconnected from timer)	
	0					
	0					
	0					
	1				OFF(output enabled and inactive level) Asynchronous: OC*x*=CCxP, OCxN=CC*x*NP; Then, if the clock exists: after a dead time *OCx=OISx* and OCxN=OISxN, assume that *OISx* and OISxN do not both correspond to active levels of *OCx* and OCxN.	
	1					
	1					
	1					

Table 0.92 Control bits for complementary output channels OCx and OCxN with break

Note: The status of the external I/O pins connected to the complementary OCx and OCxN channels depends on the OCx and OCxN channel status and GPIO registers.

8.3.1.14 Capture/Compare Enable Register 2 (PWM0_CCER2)

Address: 0xD1

Bit No.	7	6	5	4
Symbol	-	-	CC4P	CC4E
Read/Write	-	-	Read/Write	Read/Write
Po Initial Value	-	-	0	0
Bit No.	3	2	1	0

Symbol	CC3NP	CC3NE	CC3P	CC3E
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
5	CC4P	Input capture/compare 4 output polarity. Refer to the description of CC1P.
4	CC4E	Input Capture/Compare 4 output enable. Refer to the description of CC1E.
3	CC3NP	Input capture/compare 3 complementary output polarity. Refer to the description of CC1NP.
2	CC3NE	Input Capture/Compare 3 complementary output enable. Refer to the description of CC1NE.
1	CC3P	Input capture/compare 3 output polarity. Refer to the description of CC1P.
0	CC3E	Input Capture/Compare 3 output enable. Refer to the description of CC1E.

8.3.1.15 Counter High 8 bits (PWM0_CNTRH)

Address: 0xD2

Bit No.	7	6	5	4	3	2	1	0
Symbol	CNT[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CNT[15:8]	Counter high 8 bits

8.3.1.16 Counter Low 8 bits (PWM0_CNTRL)

Address: 0xD3

Bit No.	7	6	5	4	3	2	1	0
Symbol	CNT[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CNT[7:0]	Counter low 8 bits. The bit[2:0] is fixed to 0x00 when reading, i.e., the read value will have an error of 0~7 values.

8.3.1.17 Prescaler High 8 bits (PWM0_PSCRH)

Address: 0xD4

Bit No.	7	6	5	4	3	2	1	0
Symbol	PSC[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	PSC[15:8]	Prescaler High 8 bits. The prescaler is used to divide the frequency of CK_PSC. The clock frequency of the counter (f_{CK_CNT}) is equal to f_{CK_CNT}/(PSCR[15:0] + 1). The PSCR contains the value that is loaded into the current prescaler register when an update event is generated (an update event consists of the counter being cleared to 0 by the UG bit of PWM_EGR or by a slave controller in reset mode). This means that in order for the new value to take effect, an update event must be generated.

8.3.1.18 Prescaler Low 8 bits (PWM0_PSCRL)

Address: 0xD5

Bit No.	7	6	5	4	3	2	1	0
Symbol	PSC[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	PSC[7:0]	Prescaler low 8 bits. The prescaler is used to divide the frequency of CK_PSC. The clock frequency of the counter (f_{CK_CNT}) is equal to f_{CK_CNT}/(PSCR[15:0] + 1). The PSCR contains the value loaded into the current prescaler register when an update event is generated (an update event consists of the

		counter being cleared to 0 by the UG bit of PWM_EGR or by a slave controller in reset mode).This means that, in order for the new value to take effect, an update event must be generated.

8.3.1.19 Auto-reload Register High 8 bits (PWM0_ARRH)

Address: 0xD6

Bit No.	7	6	5	4	3	2	1	0
Symbol	ARR[15:8]							
Read/Write	Read/Write							
Po Initial Value	1	1	1	1	1	1	1	1

Bit No.	Symbol	Description
7~0	ARR[15:8]	Auto-reload Register Low 8 bits. The ARR contains the value that will be loaded into the actual Auto-reload Register. Please refer to the Time Base Unit for details. The counter does not work when the value of auto-reload is empty.

8.3.1.20 Auto-reload Register Low 8 bits (PWM0_ARRL)

Address: 0xD7

Bit No.	7	6	5	4	3	2	1	0
Symbol	ARR[7:0]							
Read/Write	Read/Write							
Po Initial Value	1	1	1	1	1	1	1	1

Bit No.	Symbol	Description
7~0	ARR[7:0]	Auto-reload Register Low 8 bits

8.3.1.21 Repetition Count Register (PWM0_RCR)

Address: 0xD8

Bit No.	7	6	5	4	3	2	1	0

Symbol	REP[7:0]
Read/Write	Read/Write
Po Initial Value	0

Bit No.	Symbol	Description
7~0	REP[7:0]	Repetition Count Register. With preload enabled, these bits allow the user to set the update rate of the compare registers (i.e., periodically transfer from the preload register to the current register); if an update interrupt is allowed to be generated, this affects the rate at which the update interrupt is generated as well. Each time the down counter REP_CNT reaches 0, an update event is generated and the counter REP_CNT starts counting from the REP value again. Since REP_CNT only reloads the REP value when the cycle update event U_RC occurs, the new value written to the PWM0_RCR register only takes effect when the next cycle update event occurs. This means that in the PWM mode, (REP+1) corresponds to - The number of PWM cycles in edge-aligned mode. - The number of PWM half-cycles in the center aligned mode.

8.3.1.22 Capture/Compare Register 1 High 8 bits (PWM0_CCR1H)

Address: 0xD9

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR1[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR1[15:8]	High 8-bit value of Capture/compare 1. If the CC1 channel is configured as output (CC1S bit of PWM0_CCMR1). CCR1 contains the value loaded into the current Capture/Compare 1 register (preload value). If the preload function is not selected in the PWM0_CCMR1 register (OC1PE bit), the written value is immediately passed to the current register. Otherwise this preload value is passed to the current capture/compare 1 register only when an update event occurs. The value of the current capture/compare register is compared with the value of the counter PWM0_CNT, and an output signal is

		generated on the OC1 port. If the CC1 channel is configured as input: CCR1 contains the counter value transferred by the last input capture 1 event (IC1) (this register is read-only at this time).

8.3.1.23 Capture/Compare Register 1 Low 8 bits (PWM0_CCR1L)

Address: 0xDA

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR1[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR1[7:0]	Capture/compare register 1 low 8 bits

8.3.1.24 Capture/Compare Register 2 High 8 bits (PWM0_CCR2H)

Address: 0xDB

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR2[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR2[15:8]	Capture/compare register 2 high 8 bits. If the CC2 channel is configured as output (CC2S bit of PWM0_CCMR2). CCR2 contains the value loaded into the current Capture/Compare 2 register (preload value). If the preload function is not selected in the PWM0_CCMR2 register (OC2PE bit), the written value is immediately passed to the current register. Otherwise this preload value is passed to the current capture/compare 2 register only when an update event occurs. The value of the current capture/compare register is compared with the value of the counter PWM0_CNT, and an output signal is generated on the OC2 port. If the CC2 channel is configured as input. CCR2 contains the counter value transferred by the last input capture

		2 event (IC2) (this register is read-only at this time).

8.3.1.25 Capture/Compare Register 2 Low 8 bits (PWM0_CCR2L)

Address: 0xDC

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR2[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR2[7:0]	Capture/compare register 2 low 8 bits

8.3.1.26 Capture/Compare Register 3 High 8 bits (PWM0_CCR3H)

Address: 0xDD

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR3[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR3[15:8]	Capture/compare register 3 high 8-bits If tCC3 channel is configured as output (CC3S bit of PWM0_CCMR3). CCR3 contains the value loaded into the current Capture/Compare 3 register (preload value). If the preload function is not selected in the PWM0_CCMR3 register (OC3PE bit), the written value is immediately passed to the current register. Otherwise this preload value is passed to the current capture/compare 3 register only when an update event occurs. The value of the current capture/compare register is compared with the value of the counter PWM0_CNT, and an output signal is generated on the OC3 port. If the CC3 channel is configured as input. CCR3 contains the counter value transferred by the last input capture 3 event (IC3) (this register is read-only at this time).

8.3.1.27 Capture/Compare Register 3 Low 8 bits (PWM0_CCR3L)

Address: 0xDE

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR3[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR3[7:0]	Capture/compare register 3 low 8 bits

8.3.1.28 Capture/Compare Register 4 High 8 bits (PWM0_CCR4H)

Address: 0xDF

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR4[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR4[15:8]	Capture/compare register 4 high 8-bits If CC4 channel is configured as output (CC4S bit of PWM0_CCMR4). CCR4 contains the value loaded into the current Capture/Compare 4 register (preload value). If the preload function is not selected in the PWM0_CCMR4 register (OC4PE bit), the written value is immediately passed to the current register. Otherwise this preload value is passed to the current capture/compare 4 register only when an update event occurs. The value of the current capture/compare register is compared with the value of the counter PWM0_CNT, and an output signal is generated on the OC4 port. If the CC4 channel is configured as input. CCR4 contains the counter value transferred by the last input capture 4 event (IC4) (this register is read-only at this time).

8.3.1.29 Capture/Compare Register 4 Low 8 bits (PWM0_CCR4L)

Address: 0xE2

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR4[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR4[7:0]	Capture/compare register 4 low 8 bits

8.3.1.30 Break Control Register (PWM0_BKR)

Address: 0xE3

Bit No.	7	6	5	4
Symbol	MOE	AOE	BKP	BKE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	OSSR	OSSI	LOCK	
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	0	0	0	

Bit No.	Symbol	Description
7	MOE	Main output enable. This bit is cleared to 0 asynchronously by hardware once the break input is active. According to the value set for the AOE bit, this bit can be set to 1 by software or automatically. it is only valid for channels configured as outputs. 0: disables OC and OCN outputs or forces them to idle. 1: If the corresponding enable bit (CCxE bit of PWM0_CCERx register) is set, the OC and OCN outputs are enabled. For OC and OCN enable details, refer to *Capture/Compare Enable Register 1 (PWM0_CCER1)*.
6	AOE	Auto output enable. 0: MOE can only be set to 1 by software. 1: MOE can be set to 1 by software or automatically at the next update event (if the brake input is inactive). Note: Once the LOCK level (LOCK bit in PWM0_BKR register) is set to 1, this bit cannot be modified.
5	BKR	Break input polarity.

		0: break input is active low. 1: Break input is active high. Note: Once the LOCK level (LOCK bit in PWM0_BKR register) is set to 1, this bit cannot be modified.
4	BKE	The break function enable. 0: Break input (BRK) disabled. 1: break input (BRK) enabled. Note: Once the LOCK level (LOCK bit in PWM0_BKR register) isset to 1, this bit cannot be modified.
3	OSSR	The "Off state" selection in run mode. This bit is used when MOE = 1 and the channel is a complementary output. For OC and OCN enable details, refer to *Capture/Compare Enable Register 1 (PWM0_CCER1)*. 0: when the timer is not operating, the OC/OCN output is disabled (OC/OCN enable output signal = 0). 1: When the timer is not operating, once CCxE=1 or CCxNE=1, first enable OC/OCN and output inactive level, then set OC/OCN enable output signal=1. Note: Once the LOCK level (LOCK bit in PWM0_BKR register) is set to 2, this bit cannot be modified.
2	OSSI	The "off state" selection in idle mode. This bit is used when MOE=0 and the channel is set as output. For OC and OCN enable details, refer to *Capture/Compare Enable Register 1 (PWM0_CCER1)*. 0: when the timer is not operating, the OC/OCN output is disabled (OC/OCN enable output signal = 0). 1: When the timer is not operating, once CCxE=1 or CCxNE=1, OC/OCN first outputs its idle level, and then OC/OCN enable output signal = 1. Note: Once the LOCK level (LOCK bit in PWM0_BKR register) is set to 2, this bit cannot be modified.
1~0	LOCK	Lock settings. This bit provides write protection against software errors. 00: lock off, no write protection for registers. 01: lock level 1, cannot write to the BKE, BKP, AOE bits of the PWM0_BKR register and the OIS1 bit of the PWM0_OISR register. 10: Lock level 2, where the bits in lock level 1 cannot be written, nor the CC polarity bit (the CC polarity bit is the CCxP bit of the PWM0_CCERX register once the channel in question is set as output via the CCxS bit) and the OSSR/OSSI bit. 11: Lock level 3, no writing to the bits in lock level 2, and no writing

		to the CC control bit (the CC control bit is the OCxM/OCxPE bit of the PWM0_CCMRx register once the channel in question is set to output via the CCxS bit). Note: After system reset, the LOCK bit can only be written once, and once it is written to the PWM0_BDR register, its content remains unchanged until reset.

8.3.1.31 Deadtime Register (PWM0_DTR)

Address: 0xE4

Bit No.	7	6	5	4	3	2	1	0
Symbol	UTG							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	UTG	Deadtime generator setting. These bits define the duration of the dead time between the insertion of complementary outputs. Assuming that DT represents its duration and t_{CK_PSC} is the clock pulse of PWM0. DTG[7:5]=0xx => DT=DTG[7:0]x tdtg, where tdtg=t_{CK_PSC}. (f1) DTG[7:5]=10x => DT=(64+DTG[5:0])x tdtg, where tdtg=2x t_{CK_PSC}. (f2) DTG[7:5]=110 => DT=(32+DTG[4:0])x tdtg, where tdtg=8x t_{CK_PSC} (f3) DTG[7:5]=111 => DT=(32+DTG[4:0])x tdtg, where tdtg=16x t_{CK_PSC}. (f4) For example, If t_{CK_PSC} = 125 ns (8 MHz), the possible dead time is DTG[7:0] = 0 to 7Fh, 0 to 15875 ns with a step time of 125 ns (refer to f1), DTG[7:0] = 80h to BFh , 16 µs to 31750 ns with a step time of 250 ns (refer to f2), DTG[7:0] = C0h to DFh, 32 µs to 63 µs, in increments of 1 µs (refer to f3), DTG[7:0] = E0h to FFh, 64 µs to 126 µs, in increments of 2 µs (refer to f4), Note: Once the LOCK level (LOCK bit in PWM0_BKR register) is set to 1, 2 or 3, these bits cannot be modified.

8.3.1.32 Output Idle Status Register (PWM0_OISR)

Address: 0xE5

Bit No.	7	6	5	4
Symbol	-	OIS4	OIS3N	OIS3

Read/Write	-	Read/Write	Read/Write	Read/Write
Po Initial Value	-	0	0	0
Bit No.	3	2	1	0
Symbol	OIS2N	OIS2	OIS1N	OIS1
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
6	OIS4	Output idle state 4 (OC4 output). See OIS1 bit.
5	OIS3N	Output idle state 3 (OC3N output). See OIS1N bit.
4	OIS3	Output idle state 3 (OC3 output). See OIS1 bit.
3	OIS2N	Output idle state 2 (OC2N output). See OIS1N bit.
2	OIS2	Output idle state 2 (OC2 output). See OIS1 bit.
1	OIS1N	Output idle state 1 (OC1N output). 0: when MOE=0, then after one dead time, OC1N=0. 1: When MOE=1,then after one dead time, OC1N=1. Note: After LOCK level 1, 2 or 3 has been set, this bit cannot be modified.
0	OIS1	Output idle state 1 (OC1 output). 0: when MOE = 0, OC1 = 0 after one deadtime if OC1N is enabled. 1: When MOE=0, if OC1N is enabled, then after one deadtime, OC1=1. Note: After LOCK (PWM0_BKR register) level 1, 2 or 3 has been set, this bit cannot be modified.

8.3.1.33 Capture/compare Register 5 High 8 bits (PWM0_CCR5H)

Address: 0xE8

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR5[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR5[15:8]	High 8-bit values of capture/compare register 5. CCR5 contains the value loaded into the current compare 5 register. The value of the current compare register is compared with the value of the counter PWM0_CNT, and only when the comparison matches will the delayed TRGO be output to the ADC.

8.3.1.34 Capture/compare Register 5 Low 8 bits (PWM0_CCR5L)

Address: 0xE9

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR5[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR5[7:0]	Capture/compare Register 5 Low 8 bits

8.3.1.35 Channel Direction Control Register (PWM0_DIR)

Address: 0xEA

Bit No.	7	6	5	4
Symbol	-			
Read/Write	-			
Po Initial Value	0			
Bit No.	3	2	1	0
Symbol	PWM0_CH4	PWM0_CH3	PWM0_CH2	PWM0_CH1
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7~4	--	Reserved.
3	PWM0_CH4	PWM0_CH4 direction control. 0: output; 1: input.
2	PWM0_CH3	PWM0_CH3 direction control. 0: output; 1: input.
1	PWM0_CH2	PWM0_CH2 direction control. 0: output; 1: input.
0	PWM0_CH1	PWM0_CH1 direction control. 0: output; 1: input.

8.3.1.36 Input Capture 1 Filter Register (PWM0_BKR1)

Address: 0xED

Bit No.	7	6	5	4
Symbol	-			
Read/Write	-			
Po Initial Value	-			
Bit No.	3	2	1	0
Symbol	BKINF			
Read/Write	Read/Write			
Po Initial Value	0			

Bit No.	Symbol	Description
3~0	BKINF	Input capture 1 filter. These bits define the sampling frequency of the BKIN break input and the size of the digital filter. The digital filter consists of an event counter, and the output jump is considered valid only after N events have occurred. 0000: No filter, $f_{SAMPLING}=f_{MASTER}$ 0001: $f_{SAMPLING}=f_{MASTER}$, N=2 0010: $f_{SAMPLING}=f_{MASTER}$, N=4 0011: $f_{SAMPLING}=f_{MASTER}$, N=8 0100: $f_{SAMPLING}=f_{MASTER}/2$, N=6 0101: $f_{SAMPLING}=f_{MASTER}/2$, N=8 0110: $f_{SAMPLING}=f_{MASTER}/4$, N=6 0111: $f_{SAMPLING}=f_{MASTER}/4$, N=8 1000: $f_{SAMPLING}=f_{MASTER}/8$, N=6 1001: $f_{SAMPLING}=f_{MASTER}/8$, N=8 1010: $f_{SAMPLING}=f_{MASTER}/16$, N=5 1011: $f_{SAMPLING}=f_{MASTER}/16$, N=6 1100: $f_{SAMPLING}=f_{MASTER}/16$, N=8 1101: $ff_{SAMPLING}=f_{MASTER}/32$, N=5 1110: $f_{SAMPLING}=f_{MASTER}/32$, N=6 1111: $f_{SAMPLING}=f_{MASTER}/32$, N=8

8.3.1.37 Channel Control Register (PWM0_CHX_CHXN_EN)

Address: 0xEE

Bit No.	7	6	5	4
Symbol	-	PWM0_CH3N_EN	PWM0_CH2N_EN	PWM0_CH1N_EN
Read/Write	-	Read/Write	Read/Write	Read/Write
Po Initial Value	-	1	1	1
Bit No.	3	2	1	0
Symbol	PWM0_CH4_EN	PWM0_CH3_EN	PWM0_CH2_EN	PWM0_CH1_EN
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
6	PWM0_CH3N_EN	PWM0_CH3N channel control. 0: OFF 1:ON
5	PWM0_CH2N_EN	PWM0_CH2N channel control. 0: OFF 1:ON
4	PWM0_CH1N_EN	PWM0_CH1N channel control. 0: OFF 1:ON
3	PWM0_CH4_EN	PWM0_CH4 channel control. 0: OFF 1:ON
2	PWM0_CH3_EN	PWM0_CH3 channel control. 0: OFF 1:ON
1	PWM0_CH2_EN	PWM0_CH2 channel control. 0: OFF 1:ON
0	PWM0_CH1_EN	PWM0_CH1 channel control. 0: OFF 1:ON

8.3.2 PWM1 Registers

8.3.2.1 Control Register 1 (PWM1_CR1)

Address: 0x9B

Bit No.	7	6	5	4
Symbol	ARPE	-		
Read/Write	Read/Write	-		
Po Initial Value	0	-		
Bit No.	3	2	1	0
Symbol	OPM	URS	UDIS	CEN
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	ARPE	Auto-reload preload enable bit. 0: The PWM1_ARR register has no preload register to buffer and can be written to directly. 1: PWM1_ARR register can be buffered by preload register.

3	OPM	One pulse mode. 0: the counter does not stop when an update event occurs. 1: The counter stops when the next update event (clearing the CEN bit) occurs.
2	URS	Update the request source. 0: When the update request is enabled, an update interrupt is generated whenever the register is updated. 1: When the update request is enabled, only the counter overflow generates an update interrupt.
1	UDIS	Disable Update. The software enables/disables the generation of UEV events via this bit. 0: The update event is generated whenever the counter overflows, or a software update is generated, or a hardware reset is generated via the clock/trigger mode controller. 1: No update event is generated and the shadow registers (ARR, PSC, CCRx) keep their values. If UG is set then the counter and prescaler are reinitialized.
0	CEN	Enable counter. 0: disabled. 1: enabled.

8.3.2.2 Control Register 2 (PWM1_CR2)

Address: 0x9C

Bit No.	7	6	5	4
Symbol	TI1S	MMS		
Read/Write	Read/Write	Read/Write		
Po Initial Value	0	0		
Bit No.	3	2	1	0
Symbol	-	-	-	-
Read/Write	-	-	-	-
Po Initial Value	-	-	-	-

Bit No.	Symbol	Description
7	TI1S	TI1 selection. 0: CC1 input pin connected to TI1 (input of the digital filter). 1: CC1, CC2 and CC3 pins are connected to TI1 after XOR.

6~4	MMS	Master mode selection. These 3 bits are used to select the synchronization information (TRGO) sent to PWM0 in master mode. The possible combinations are as follows: 000: Reset. The UG bit of the PWM1_EGR register is used as the trigger output (TRGO). If the trigger input (from the mode controller in reset mode) generates a reset, there is a delay in the signal on TRGO relative to the actual reset. 001: Enable. The counter enable signal CNT_EN is used as a trigger output (TRGO). It is used to start multiple timers at the same time or to control the enable of slave timers over a period of time. The counter enable signal is generated by the logical or of the trigger input signal in CEN control bit and gated mode. When the counter enable signal is controlled by the trigger input, there is a delay on TRGO unless master/slave mode is selected (see the description of the MSM bit in the PWM1_SMCR register). 010: Update. The update event is selected as trigger output. 011: Reserved. 100: Reserved. 101: Compare. OC1REF signal is used as the trigger output (TRGO). 110: Compare. OC2REF signal is used as the trigger output (TRGO). 111: Compare. OC3REF signal is used as the trigger output (TRGO).

8.3.2.3 Slave Mode Control Register (PWM1_SMCR)

Address: 0x9D

Bit No.	7	6	5	4
Symbol	MSM	TS		
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	0	0	0	
Bit No.	3	2	1	0
Symbol	-	SMS		
Read/Write	-	Read/Write		
Po Initial Value	-	0		

Bit No.	Symbol	Description
7	MSM	Master/Slave Mode. 0: no effect.

		1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the timer (via TRGO) and its slave timer.
6~4	TS	Trigger selection. For selecting the trigger input of synchronous counter. 000: Reserved 001: Reserved 010: Reserved 011: Internal trigger ITR connected to PWM0 TRGO 100: Edge detector for TI1 (TI1F_ED) 101: Timer input after filtering (TI1FP1) 110: Timer input after filtering (TI2FP2) 111: Reserved Note: These bits can only be changed when they are not used (e.g. SMS=000) to avoid false edge detection when they are changed.
2~0	SMS	Clock/trigger/slave mode selection. When an external signal is selected, the active edge of the trigger signal (TRGI) is related to the polarity of the selected external input (see Description of the Input Control Register and Control Register) 000: Clock/trigger controller disabled. If CEN = 1, the prescaler is driven directly by the internal clock. 001: Encoder mode 1. Depending on the level of TI1FP1, the counter counts up/down on the edge of TI2FP2. 010: Encoder mode 2. Depending on the level of TI2FP2, the counter counts up/down on the edge of TI1FP1. 011: Encoder mode 3. Depending on the level of the other input, the counter counts up/down on the edges of TI1FP1 and TI2FP2. 100: Reset mode. The counter is reinitialized on the rising edge of the selected trigger input (TRGI) and a signal is generated to update the register. 101: Gated mode. When the trigger input (TRGI) is high, the counter is clocked on. Once the trigger input becomes low, the counter stops (but does not reset). The started and stopped of counter is in controlled. 110: Trigger mode. The counter is started (but not reset) on the rising edge of the trigger input TRGI, and only the start of the counter is in controlled. 111: External clock mode 1. The rising edge of the selected trigger input (TRGI) drives the counter.

8.3.2.4 Interrupt Enable Register (PWM1_IER)

Address: 0x9E

Bit No.	7	6	5	4
Symbol	-	TIE	-	-
Read/Write	-	Read/Write	-	-
Po Initial Value	-	0	-	-
Bit No.	3	2	1	0
Symbol	CC3IE	CC2IE	CC1IE	UIE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
6	TIE	Trigger Interrupt enable. 0: Disabled 1: enabled
3	CC3IE	Enable capture/compare 3 interrupt. 0: Disabled 1: enabled
2	CC2IE	Enable capture/compare 2 interrupt. 0: Disabled 1: enabled
1	CC1IE	Enable capture/compare 1 interrupt. 0: Disabled 1: enabled
0	UIE	Enable update interrupt. 0: Disabled 1: enabled

8.3.2.5 Status Register 1 (PWM1_SR1)

Address: 0x9F

Bit No.	7	6	5	4
Symbol	-	TIF	-	-
Read/Write	-	Read/Write	-	-
Po Initial Value	-	0	-	-
Bit No.	3	2	1	0

Symbol	CC3IF	CC2IF	CC1IF	UIF
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
6	TIF	Trigger interrupt flag. When a trigger event occurs (a active edge is detected at the TRGI input when the slave mode controller is in a mode other than gated mode. Or any edge in gated mode), it is set to 1 by hardware. it is written/cleared to 0 by software. 0: no trigger events occur. 1: Trigger an interrupt to wait for response.
3	CC3IF	Capture/compare 3 interrupt flag. Refer to CC1IF description.
2	CC2IF	Capture/compare 2 interrupt flag. Refer to CC1IF description.
1	CC1IF	Capture/compare 1 interrupt flag. If channel CC1 is configured as output mode: This bit is set to 1 by hardware when the counter value matches the comparison value and is cleared to 0 by software. 0: no match occurs. 1: The value of PWM1_CNT matches the value of PWM1_CCR1. If channel CC1 is configured as input mode: This bit is set to 1 by hardware when a capture event occurs, and it is cleared to 0 by software or by reading PWM1_CCR1L. 0: no input capture generated. 1: The counter value has been captured (copied) to PWM1_CCR1 (an edge with the same polarity as the selected one is detected on IC1).
0	UIF	Update interrupt flag. This bit is set to 1 by hardware when an update event is generated. it is cleared to 0 by software. 0: no update events generated. 1: Update event wait accordingly. This bit is set to 1 by hardware when the register is updated, -- If UDIS=0 in the PWM1_CR1 register, the counter overflows. -- if UDIS=0 and URS=0 of the PWM1_CR1 register when setting the UG bit of the PWM1_EGR register software to reinitialize the counter CNT.

8.3.2.6 Status Register 2 (PWM1_SR2)

Address: 0xA1

Bit No.	7	6	5	4
Symbol	-			
Read/Write	-			
Po Initial Value	-			
Bit No.	3	2	1	0
Symbol	CC3OF	CC2OF	CC1OF	-
Read/Write	Read/Write	Read/Write	Read/Write	-
Po Initial Value	0	0	0	-

Bit No.	Symbol	Description
3	CC3OF	Capture/compare 3 repetition capture flag. See CC1OF description.
2	CC2OF	Capture/compare 2 repetition capture flag. See CC1OF description.
1	CC1OF	Capture/Compare 1 repetition capture flag. This flag can be set to 1 by hardware only when the corresponding channel is configured for input capture. writing 0 to clear the bit. 0: no repetition capture is generated. 1: The value of the counter is captured to the PWM1_CCR1 register when the status of CC1IF is already 1.

8.3.2.7 Event Generation Register (PWM1_EGR)

Address: 0xA2

Bit No.	7	6	5	4
Symbol	-	TG	-	-
Read/Write	-	Read/Write	-	-
Po Initial Value	-	0	-	-
Bit No.	3	2	1	0
Symbol	CC3G	CC2G	CC1G	UG
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
6	TG	Generate trigger event. This bit is set to 1 by software to generate a trigger event that is automatically cleared to 0 by hardware. 0: no behavior. 1: TIF=1 of PWM1_SR1 register, if the corresponding interrupt is turned on (TIE=1), the corresponding interrupt will be generated.
3	CC3G	Generate capture/compare 3 event. Refer to CC1G description.
2	CC2G	Generate capture/compare 2 event. Refer to CC1G description.
1	CC1G	Generate capture/compare 1 event. This bit is set to 1 by software to generate a capture/compare event, which is automatically cleared to 0 by hardware. 0: no behavior. 1: Generate a capture/compare event on channel CC1. If channel CC1 is configured as output: Set CC1IF=1 to generate the corresponding interrupt if the corresponding interrupt is turned on. If channel CC1 is configured as input: The current counter value is captured to the PWM1_CCR1 register, set CC1IF=1, and if the corresponding interrupt is turned on, the corresponding interrupt is generated. If CC1IF is already 1, then set CC1OF=1.
0	UG	Generates an update event. This bit is set to 1 by software and automatically cleared to 0 by hardware. 0: no behavior. 1: Reinitialize the counter and generate an update event. Note that the counter of prescaler is also cleared to 0.

8.3.2.8 Capture/Compare Mode Register 1 (PWM1_CCMR1)

The channel can be used in input (capture mode) or output (compare mode), and the direction of the channel is defined by the corresponding CC1S bit. The other bits of the register have different functions in input and output mode. OCxx describes the function of the channel in output mode and ICxx describes the function of the channel in input mode. Therefore it must be noted that the same bit has a different function in output mode and in input mode.

Address: 0xA3

- Channel configured as input capture mode

Bit No.	7	6	5	4
Symbol	IC1F			
Read/Write	Read/Write			
Po Initial Value	0			
Bit No.	3	2	1	0
Symbol	IC1PSC		CC1S	
Read/Write	Read/Write		Read/Write	
Po Initial Value	0		0	

Bit No.	Symbol	Description
7~4	IC1F	Input capture 1 filter. These bits define the sampling frequency of the TI1 input and the size of the digital filter. The digital filter consists of an event counter, and output jump is considered valid only after N events have occurred. 0000: no filter, $f_{SAMPLING}=f_{MASTER}$ 0001: $f_{SAMPLING}=f_{MASTER}$, N=2 0010: $f_{SAMPLING}=f_{MASTER}$, N=4 0011: $f_{SAMPLING}=f_{MASTER}$, N=8 0100: $f_{SAMPLING}=f_{MASTER}/2$, N=6 0101: $f_{SAMPLING}=f_{MASTER}/2$, N=8 0110: $f_{SAMPLING}=f_{MASTER}/4$, N=6 0111: $f_{SAMPLING}=f_{MASTER}/4$, N=8 1000: $f_{SAMPLING}=f_{MASTER}/8$, N=6 1001: $f_{SAMPLING}=f_{MASTER}/8$, N=8 1010: $f_{SAMPLING}=f_{MASTER}/16$, N=5 1011: $f_{SAMPLING}=f_{MASTER}/16$, N=6 1100: $f_{SAMPLING}=f_{MASTER}/16$, N=8 1101: $f_{SAMPLING}=f_{MASTER}/32$, N=5 1110: $f_{SAMPLING}=f_{MASTER}/32$, N=6 1111: $f_{SAMPLING}=f_{MASTER}/32$, N=8 Note: Even for channels with complementary outputs, this bit field is non-preloaded and does not take into account the value of the CCPC (PWM1_CR2 register).
3~2	IC1PSC	Input/capture 1 prescaler. These 2 bits define the prescaler factor of the CC1 input (IC1). Once CC1E=0 (in PWM1_CCER register), the prescaler is reset. 00: without prescaler, one capture is triggered for each edge detected on the capture input. 01: capture is triggered every 2 events.

		10: capture is triggered every 4 events. 11: Capture is triggered every 8 events. Note: The internal event counter is not reset when IC1PSC is dynamically changed. In this case, the old value is used for all captures when CC1E=1. If you want to use the new value immediately, you can clear the CC1E bit and set it again.
1~0	CC1S	Capture/Compare 1 Select. These 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: CC1 channel is configured as output. 01: CC1 channel is configured as input and IC1 is mapped to TI1FP1. 10: CC1 channel is configured as input and IC1 is mapped to TI2FP1. 11: CC1 channel is configured as input and IC1 is mapped to TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM1_SMCR register). Note: CC1S can be written only when the channel is off (CC1E=0 of PWM1_CCER1 register).

- Channel configured as output compare mode

Bit No.	7	6	5	4
Symbol	-	OC1M		
Read/Write	-	Read/Write		
Po Initial Value	-	0		
Bit No.	3	2	1	0
Symbol	OC1PE	-	CC1S	
Read/Write	Read/Write	-	Read/Write	
Po Initial Value	0	-	0	

Bit No.	Symbol	Description
7	--	Reserved.
6~4	OC1M	Output compare 1 mode. These 3 bits define the action of the output reference signal OC1REF, which determines the value of OC1. OC1REF is active high, while the active level of OC1 depends on the CC1P bit. 000: Frozen. Comparison between the output compare register PWM1_CCR1 and the counter PWM1_CNT does not work for OC1REF. 001: Set the output of channel 1 to active level when matching.

		Forced OC1REF to high when the value of counter PWM1_CNT is the same as capture/compare register 1 (PWM1_CCR1). 010: Set the output of channel 1 to inactive level when matched. Forced OC1REF to low when the value of counter PWM1_CNT is the same as capture/compare register 1 (PWM1_CCR1). 011: Reverse. Reverse the level of OC1REF when PWM1_CCR1= PWM1_CNT. 100: Forced to inactive level. Forces OC1REF at low. 101: Forced to active level. Force OC1REF at high. 110: PWM Mode 1- In up-counting mode, channel 1 is active once PWM1_CNT< PWM1_CCR1, otherwise it is invalid. 111: PWM mode 2- In up-counting mode, channel 1 is invalid once PWM1_CNT< PWM1_CCR1, otherwise it is valid. Note: In PWM mode 1 or PWM mode 2, the OC1REF level changes only when the comparison result is changed or when switching from frozen mode to PWM mode in the output compare mode (refer to 8.2.4.5 *PWM mode* for details)
3	OC1PE	Output Compare 1 Preload Enable. 0: Preload function of PWM1_CCR1 register is disabled, PWM1_CCR1 register can be written at any time, and the newly written value takes effect immediately. 1: Enable the preload function of PWM1_CCR1 register, the read/write operation only operates on the preload register, and the preload value of PWM1_CCR1 is loaded into the current register when the update event comes. Note: In order to operate correctly, the preload function must be enabled in PWM mode. However, in one pulse mode (PWM1_CR1 register with OPM=1), it is not necessary.
2	--	Reserved.
1~0	CC1S	CC1S[1:0]: Capture/Compare 1 Selection. These 2 bits define the direction of the channel (input/output), and the selection of the input pin. 00: CC1 channel is configured as output. 01: CC1 channel is configured as input and IC1 is mapped to TI1FP1. 10: CC1 channel is configured as input and IC1 is mapped to TI2FP1. 11: CC1 channel is configured as input and IC1 is mapped to TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM1_SMCR register). Note: CC1S can be written only when the channel is off (CC1E=0 of PWM1_CCER1 register).

8.3.2.9 Capture/Compare Mode Register 2 (PWM1_CCMR2)

Refer to *Capture/Compare Mode Register 1 (PWM1_CCMR1)* for more details.

Address: 0xA4

- **Input Capture Mode**

Bit No.	7	6	5	4
Symbol	IC2F			
Read/Write	Read/Write			
Po Initial Value	0			
Bit No.	3	2	1	0
Symbol	IC2PSC		CC2S	
Read/Write	Read/Write		Read/Write	
Po Initial Value	0		0	

Bit No.	Symbol	Description
7~4	IC2F	Input capture 2 filter.
3~2	IC2PSC	Input capture 2 prescaler.
1~0	CC2S	Capture/Compare 2 selection. This bit defines the direction of the channel (I/O), and the selection of the input pin: 00: CC2 channel is configured as output. 01: CC2 channel is configured as input and IC2 is mapped to TI2FP2. 10: CC2 channel is configured as input and IC2 is mapped to TI1FP2. 11: CC2 channel is configured as input and IC2 is mapped to TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM1_SMCR register). Note: CC2S can be written only when the channel is off (CC2E=0 of PWM1_CCER1 register).

- Output Compare Mode

Bit No.	7	6	5	4
Symbol	-	OC2M		
Read/Write	-	Read/Write		
Po Initial	-	0		

Value				
Bit No.	3	2	1	0
Symbol	OC2PE	-	CC2S	
Read/Write	Read/Write	-	Read/Write	
Po Initial Value	0	-	0	

Bit No.	Symbol	Description
6~4	OC2M	Output compare 2 mode.
3	OC2PE	Output compare 2 prescaler enable.
1~0	CC2S	Capture/Compare 2 selection. This bit defines the direction of the channel (I/O), and the selection of the input pin. 00: CC2 channel is configured as output. 01: CC2 channel is configured as input and IC2 is mapped toTI2FP2. 10: CC2 channel is configured as input and IC2 is mapped to TI1FP2. 11: CC2 channel is configured as input and IC2 is mapped to TRC. This mode works only when the internal trigger input is selected (selected by the TS bit of the PWM1_SMCR register). Note: CC2S can be written only when the channel is off (CC2E=0 of PWM1_CCER1 register).

8.3.2.10 Capture/Compare Mode Register 3 (PWM1_CCMR3)

Refer to *Capture/Compare Mode Register 1 (PWM1_CCMR1)* for more details.

Address: 0xA5

- Input capture mode

Bit No.	7	6	5	4
Symbol	IC3F			
Read/Write	Read/Write			
Po Initial Value	0			
Bit No.	3	2	1	0
Symbol	IC3PSC		CC3S	
Read/Write	Read/Write		Read/Write	
Po Initial Value	0		0	

Bit No.	Symbol	Description
7~4	IC3F	Input capture 3 filter.
3~2	IC3PSC	Input capture 3 prescaler.
1~0	CC3S	Capture/Compare 3 Selection. These 2 bits define the direction of the channel (I/O) and the selection of input pin: 00: CC3 channel is configured as output. 01: CC3 channel is configured as input, IC3 is mapped to TI3FP3. 10: Reserved 11: Reserved Note: CC3S can be written only when the channel is off (CC3E=0 of PWM1_CCER2 register).

- Output compare mode

Bit No.	7	6	5	4
Symbol	-	OC3M		
Read/Write	-	Read/Write		
Po Initial Value	-	0		
Bit No.	3	2	1	0
Symbol	OC3PE	-	CC3S	
Read/Write	Read/Write	-	Read/Write	
Po Initial Value	0	-	0	

Bit No.	Symbol	Description
6~4	OC3M	Output compare 3 mode.
3	OC3PE	Output compare 3 prescaler enable.
1~0	CC3S	Capture/compare 3 selection. This bit defines the direction of the channel (input/output), and the selection of the input pin. 00: CC3 channel is configured as output. 01: CC3 channel is configured as input and IC3 is mapped to TI3FP3. 10: Reserved 11: Reserved Note: CC3S can be written only when CC3E=0) of PWM1_CCER2 register when the channel is off.

8.3.2.11 Capture/Compare Enable Register 1 (PWM1_CCER1)

Address: 0xA6

Bit No.	7	6	5	4
Symbol	-	-	CC2P	CC2E
Read/Write	-	-	Read/Write	Read/Write
Po Initial Value	-	-	0	0
Bit No.	3	2	1	0
Symbol	-	-	CC1P	CC1E
Read/Write	-	-	Read/Write	Read/Write
Po Initial Value	-	-	0	0

Bit No.	Symbol	Description
5	CC2P	Input capture/compare 2 output polarity. Refer to the description of CC1P.
4	CC2E	Input Capture/Compare 2 output enable. Refer to the description of CC1E.
1	CC1P	Input Capture/Compare 1 Output polarity. The CC1 channel is configured as output: 0: OC1 is active high. 1: OC1 is active low. The CC1 channel is configured as input or capture (Refer to Figure8.41). 0: Captures occurs on the rising edge of TI1F or TI2F. 1: Capture occurs on the falling edge of TI1F or TI2F.
0	CC1E	Input Capture/Compare 1 output enable. The CC1 channel is configured as output. 0: Off- OC1 disables output. 1: On - OC1 signal output to the corresponding output pin. The CC1 channel is configured as input: This bit determines whether the counter value can be captured into the PWM1_CCR1 register: 0: Capture disabled. 1: Capture enabled.

8.3.2.12 Capture/Compare Enable Register 2 (PWM1_CCER2)

Address: 0xA7

Bit No.	7	6	5	4
Symbol	-	-	-	-
Read/Write	-	-	-	-
Po Initial Value	-	-	-	-
Bit No.	3	2	1	0
Symbol	-	-	CC3P	CC3E
Read/Write	-	-	Read/Write	Read/Write
Po Initial Value	-	-	0	0

Bit No.	Symbol	Description
1	CC3P	Input capture/compare 3 output polarity. Refer to the description of CC1P.
0	CC3E	Input Capture/Compare 3 output enable. Refer to the description of CC1E.

8.3.2.13 Counter High 8 bits (PWM1_CNTRH)

Address: 0xA9

Bit No.	7	6	5	4	3	2	1	0
Symbol	CNT[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CNT[15:8]	Counter high 8 bits

8.3.2.14 Counter Low 8 bits (PWM1_CNTRL)

Address: 0xAA

Bit No.	7	6	5	4	3	2	1	0
Symbol	CNT[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CNT[7:0]	Counter low 8 bits.

8.3.2.15 Prescaler Register (PWM1_PSCR)

Address: 0xAB

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	PSC			
Read/Write	-	-	-	-	Read/Write			
Po Initial Value	-	-	-	-	0			

Bit No.	Symbol	Description
3~0	PSC	The value of the prescaler. The prescaler is used to divide the frequency of CK_PSC. The clock frequency of the counter (f_{CK_CNT}) is equal to f_{CK_PSC}/2^(PSC[3:0]) and is cleared to 0 by hardware. The PSCR contains the value that is loaded into the current prescaler register when an update event is generated (the update event consists of the counter being cleared by the UG bit of PWM_EGR). This means that in order for the new value to take effect, an update event must be generated.

8.3.2.16 Auto-reload Register High 8 bits (PWM1_ARRH)

Address: 0xAC

Bit No.	7	6	5	4	3	2	1	0
Symbol	ARR[15:8]							
Read/Write	Read/Write							
Po Initial Value	1	1	1	1	1	1	1	1

Bit No.	Symbol	Description
7~0	ARR[15:8]	High octet value of Auto-reload. The ARR contains the value that will be loaded into the actual Auto-reload Register. For details, please refer to *Time Base Unit*. The counter does not work when the value of auto-reload is empty.

8.4.2.17 Auto-reload Register Low 8 bits (PWM1_ARRL)

Address: 0xAD

Bit No.	7	6	5	4	3	2	1	0
Symbol	ARR[7:0]							
Read/Write	Read/Write							
Po Initial Value	1	1	1	1	1	1	1	1

Bit No.	Symbol	Description
7~0	ARR[7:0]	Auto-reload Register Low 8 bits

8.3.2.18 Capture/Compare Register 1 High 8 bits (PWM1_CCR1H)

Address: 0xAE

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR1[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR1[15:8]	Capture/compare 1 high 8-bit value. If the CC1 channel is configured as output the CC1S bit of PWM1_CCMR1. CCR1 contains the value loaded into the current Capture/Compare 1 register (preload value). If the preload function is not selected in the PWM1_CCMR1 register (OC1PE bit), the written value is immediately passed to the current register. Otherwise this preload value is passed to the current capture/compare 1 register only when an update event occurs. The value of the current capture/compare register is compared with the value of the counter PWM1_CNT, and an output signal is generated on the OC1 port. If the CC1 channel is configured as input: CCR1 contains the counter value at the time of the last input capture 1 event (IC1) (this register is read-only at this time).

8.3.2.19 Capture/Compare Register 1 Low 8 bits (PWM1_CCR1L)

Address: 0xAF

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR1[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR1[7:0]	Capture/Compare Register 1 Low 8 bits

8.3.2.20 Capture/Compare Register 2 High 8 bits (PWM1_CCR2H)

Address: 0xB1

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR2[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR2[15:8]	Capture/compare 2 high 8 bits value. If the CC2 channel is configured as output (CC2S bit of PWM1_CCMR2). CCR2 contains the value loaded into the current Capture/Compare 2 register (preload value). If the preload function is not selected in the PWM1_CCMR2 register (OC2PE bit), the written value is passed to the current register immediately. Otherwise this preload value is passed to the current capture/compare 2 register only when an update event occurs. The value of the current capture/compare register is compared with the value of the counter PWM1_CNT, and an output signal is generated on the OC2 port. If the CC2 channel is configured as input: CCR2 contains the counter value transferred by the last input capture 2 event (IC2) (this register is read-only at this time).

8.3.2.21 Capture/Compare Register 2 Low 8 bits (PWM1_CCR2L)

Address: 0xB2

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR2[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR2[7:0]	Capture/Compare Register 2 Low 8 bits

8.3.2.22 Capture/Compare Register 3 High 8 bits (PWM1_CCR3H)

Address: 0xB3

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR3[15:8]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR3[15:8]	Capture/compare 3 high 8 bits value. If the CC3 channel is configured as output (CC3S bit of PWM1_CCMR3). CCR3 contains the value loaded into the current Capture/Compare 3 register (preload value). If the preload function is not selected in the PWM1_CCMR3 register (OC3PE bit), the written value is immediately passed to the current register. Otherwise this preload value is passed to the current capture/compare 3 register only when an update event occurs. The value of the current capture/compare register is compared with the value of the counter PWM1_CNT, and an output signal is generated on the OC3 port. If the CC3 channel is configured as input. CCR3 contains the counter value transferred by the last input capture 3 event (IC3) (this register is read-only at this time).

8.3.2.23 Capture/Compare Register 3 Low 8 bits (PWM1_CCR3L)

Address: 0xB4

Bit No.	7	6	5	4	3	2	1	0
Symbol	CCR3[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CCR3[7:0]	Capture/Compare Register 3 Low 8 bits

8.3.2.24 PWM1 Channel Control Register (PWM1_DIR)

Address: 0xEB

Bit No.	7	6	5	4
Symbol	-	-	-	-
Read/Write	-	-	-	-
Po Initial Value	-	-	-	-
Bit No.	3	2	1	0
Symbol	-	PWM1_CH3	PWM1_CH2	PWM1_CH1
Read/Write	-	Read/Write	Read/Write	Read/Write
Po Initial Value	-	0	0	0

Bit No.	Symbol	Description
7~3	--	Reserved
2	PWM1_CH3	PWM1_CH3 Direction control. 0: output; 1:input
1	PWM1_CH2	PWM1_CH2 Direction control. 0: output; 1:input
0	PWM1_CH1	PWM1_CH1 Direction control. 0: output; 1:input

8.3.3 PWM0/1 Registers

8.3.3.1 Status Register 3 (PWM_SR3)

Address: 0xEC

Bit No.	7	6	5	4
Symbol	-	PWM1_CC3F	PWM1_CC2F	PWM1_CC1F
Read/Write	-	Read	Read	Read

Po Initial Value	-			0
Bit No.	3	2	1	0
Symbol	PWM0_CC4F	PWM0_CC3F	PWM0_CC2F	PWM0_CC1F
Read/Write	Read	Read	Read	Read
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
6	PWM1_CC3F	PWM1 Module: The default value is 0. See PWM0_CC1F bit.
5	PWM1_CC2F	PWM1 Module: The default value is 0. See PWM0_CC1F bit.
4	PWM1_CC1F	PWM1 Module: The default value is 0. See PWM0_CC1F bit.
3	PWM0_CC4F	PWM0 Module: The default value is 0. See PWM0_CC1F bit.
2	PWM0_CC3F	PWM0 Module: The default value is 0. See PWM0_CC1F bit.
1	PWM0_CC2F	PWM0 Module: The default value is 0. See PWM0_CC1F bit.
0	PWM0_CC1F	PWM0Module: The default value is 0. 0: The bit is set to 0 by hardware when a capture event occurs and is captured on the falling edge of the CC1 input signal. 1: The bit is set to 1 by hardware when a capture event occurs and is captured on the rising edge of the CC1 input signal. Note: When TRGO of PWM1 is captured as TRC data, there is no effect on this bit.

8.3.3.2 PWM IO Selection Register 0 (PWM_IO_SEL0)

XRAM_SFR Address: 0x2047

Bit No.	7	6	5	4
Symbol	PWM_ETR_EN	PWM_ETR	PWM0_CH3_SEL	
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	0	0	0	
Bit No.	3	2	1	0
Symbol	PWM0_CH2_SEL		PWM0_CH1_SEL	
Read/Write	Read/Write		Read/Write	
Po Initial Value	0		0	

Bit No.	Symbol	Description
7	PWM_ETR_EN	PWM_ETR IO enable. 1: select ETR 0: do not select ETR

6	PWM_ETR	PWM_ETR IO selection. 0: PA3. 1: PD4
5~4	PWM0_CH3_SEL	PWM0_CH3 IO selection. 0: PD4. 1: PB2. 2: PC4
3~2	PWM0_CH2_SEL	PWM0_CH2 IO selection. 0: PB0. 1: PB6. 2: PC6
1~0	PWM0_CH1_SEL	PWM0_CH1 IO selection. 0: PB1. 1: PB7. 2: PC7

8.3.3.3 PWM IO Selection Register 1 (PWM_IO_SEL1)

xram_sfr Address: 0x2048

Bit No.	7	6	5	4
Symbol	PWM_BRK_EN	PWM_BRK		PWM0_CH3N_SEL
Read/Write	Read/Write	Read/Write		Read/Write
Po Initial Value	0	0		0
Bit No.	3	2	1	0
Symbol	PWM0_CH2N_SEL	PWM0_CH1N_SEL	PWM1_CH2_SEL	PWM1_CH1_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	PWM_BRK_EN	PWM_BRK IO Enable 1: Selected BRK 0: Do not select BRK
6~5	PWM_BRK	PWM_BRK IO Selection 0: PA1; 1: PB0; 2: PC6
4	PWM0_CH3N_SEL	PWM0_CH3N IO Selection 0: PA5; 1: PD3
3	PWM0_CH2N_SEL	PWM0_CH2N IO Selection 0: PA4; 1: PD2
2	PWM0_CH1N_SEL	PWM0_CH1N IO Selection 0: PD1; 1: PB5
1	PWM1_CH2_SEL	PWM1_CH2 IO Selection 0: PD1; 1: PC2
0	PWM1_CH1_SEL	PWM1_CH1 IO Selection 0: PA3; 1: PD0

8.3.3.4 PWM Clock Configuration Register (PWM_SYS_CFG)

XRAM_SFR Address: 0x205E

Bit No.	7	6	5	4
Symbol	-			
Read/Write	-			
Po Initial Value	-			
Bit No.	3	2	1	0
Symbol	PWM1_EN	PWM0_EN	PWM1_CLK_SEL	PWM0_CLK_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
3	PWM1_EN	PWM1 module enable control 1: Enabled 0: Disabled. Counter is reset and clock is turned off.
2	PWM0_EN	PWM0 module enable control. 1: Enabled. 0: Disabled. Counter is reset and clock is turned off.
1	PWM1_CLK_SEL	PWM1 Clock selection register. 0: Selected PLL_24Mhz clock 1: Selected XRAL
0	PWM0_CLK_SEL	PWM0Clock selection register. 0: Selected PLL_24Mhz clock. 1: Selected XRAL

8.4 PWM Configuration Procedures

8.4.1 PWM0 Register Configuration Procedures

Configuration process.

1) Configure PWM0_EN=1 (operate enable) in the PWM_SYS_CFG register; configure the PWM0_DIR register to select the direction of the PWM channel.
2) Configure PWM0_CR1, PWM0_PSCRH, PWM0_PSCRL, PWM0_ARRH, PWM0_ARRL and PWM0_RCR to select the count mode.
3) If in output compare mode, configure the PWM0_CCR1H, PWM0_CCR1L PWM0_CCR2H, PWM0_CCR2L, PWM0_CCR3H, PWM0_CCR3L, PWM0_CCR4H, and PWM0_CCR4L registers to configure the compare values.
4) If with preload enable, configure the UG bit of the PWM0_EGR register to update the shadow

register.

5) Configure PWM0_IER to set interrupt enable.
6) Configure PWM0_CCMR1, PWM0_CCMR2, PWM0_CCMR3, PWM0_CCMR4, to select input capture or output compare modes and the parameters in the corresponding modes.
7) Configure the PWM0_CCER1 and PWM0_CCER2 registers to enable the capture or output compare and to select the polarity of the channel.
8) The COMG bit of the PWM0_EGR register can be configured to generate COM update events such as OCxM, CCxE, CCxP, CCxNE, CCxNP if preload enable is available.
9) Configure the PWM0_ETR, PWM0_CR2, PWM0_SMCR registers to select the trigger mode.
10) Configure PWM0_DTR to select the dead time, PWM0_OISR to select the idle level of the channel.
11) Configure PWM0_BKR register to set the MOE bit and select the break and polarity.
12) Configure CEN=1 of PWM0_CR1 register to start the counter.

Configuration constraints.

1) Input signal transformation cannot be too fast when input capture, and output waveform transformation cannot be too short, i.e. if the interval between two interrupts is too short, the system interrupt can not handle, and may miss the interrupt.
2) It is not recommended to configure PWM0_CNTRH and PWM0_CNTL registers during the counting period.
3) If configured as input capture mode, it requires two steps: first, configure the CCxS bit in the PWM0_CCMRx register to select the mapping in input mode; second, configure the ICxF and ICxPSC in the PWM0_CCMRx register to select the filter and prescaler factors.
4) Note that when reading PWM0_CR1 immediately after writing the PWM0_CR1 register, there is a problem with the read value. It is necessary to delay the reading for some time, then the read value is correct. This is because the CEN and DIR bits in the PWM0_CR1 register are processed twice across the time domain, then the CEN and DIR bits are the correct read values.

8.4.2 PWM1 Register Configuration Procedures

Configuration standard process.

1) Configure operate enable bit PWM1_EN=1 in the PWM_SYS_CFG register; configure the PWM1_DIR register to select the direction of the PWM channel.
2) Configure PWM1_CR1, PWM1_PSCR, PWM1_ARRH and PWM1_ARRL to select the counting mode.
3) If in output compare mode, configure PWM1_CCR1H PWM1_CCR1L, PWM1_CCR2H, PWM1_CCR2L, PWM1_CCR3H and PWM1_CCR3L registers to configure the compare values.
4) If with preload enable, configure the UG bit of the PWM1_EGR register to update the shadow register.
5) Configure PWM1_IER to set interrupt enable.
6) Configure PWM1_CCMR1, PWM1_CCMR2, PWM1_CCMR3 to select the input capture or output compare modes and the parameters in the corresponding modes.

7) Configure PWM1_CCER1 and PWM1_CCER2 registers to enable the capture or output compare and to select the polarity of the channel.
8) Configure PWM1_CR2 and PWM1_SMCR registers to select the trigger mode.
9) Configure CEN=1 of PWM1_CR1 register to start the counter.

Configuration constraints.

1) Input signal transformation during input capture cannot be too fast, and output waveform transformation cannot be too short, i.e. the interval between two interrupts is too short, the system interrupt can not handle, and may miss the interrupt.

2) It is not recommended to configure the PWM1_CNTRH, PWM1_CNTL registers during the counting period.

3) If configured as input capture mode, it requires two steps: first, configure the CCxS bit in the PWM1_CCMRx register to select the mapping in input mode; second, configure the ICxF and ICxPSC in the PWM1_CCMRx register to select the filter and prescaler factors.

4) Note that when reading PWM1_CR1 immediately after writing the PWM1_CR1 register, there is a problem with the read value. It is necessary to delay the reading for some time, then the read value is correct, because the CEN and DIR bits in the PWM1_CR1 register are processed twice across the time domain, until then CEN and DIR bits are the read values.

9 ADC

9.1 Introduction

ADC can perform analog-to-digital conversion for any channel selected by the software. The reference voltage of ADC is connected to the VCC of the chip. All the ADC channels can input independent analog signals. Each time ADC module converts one channel, ADC_START=0→1 turns on the conversion, and updates the ADC result register and generates an interrupt when the conversion is complete.

The ADC module of BS9000AMxx has the following features:

- Single-ended, 12-bit resolution linear successive-approximation ADC
- Single conversion mode
- Sampling time and conversion speed configurable
- Support wake-up in wait mode
- 26 analog input channels (including 1 test channel), each of which can be decided whether to support ADC function by configuration byte.

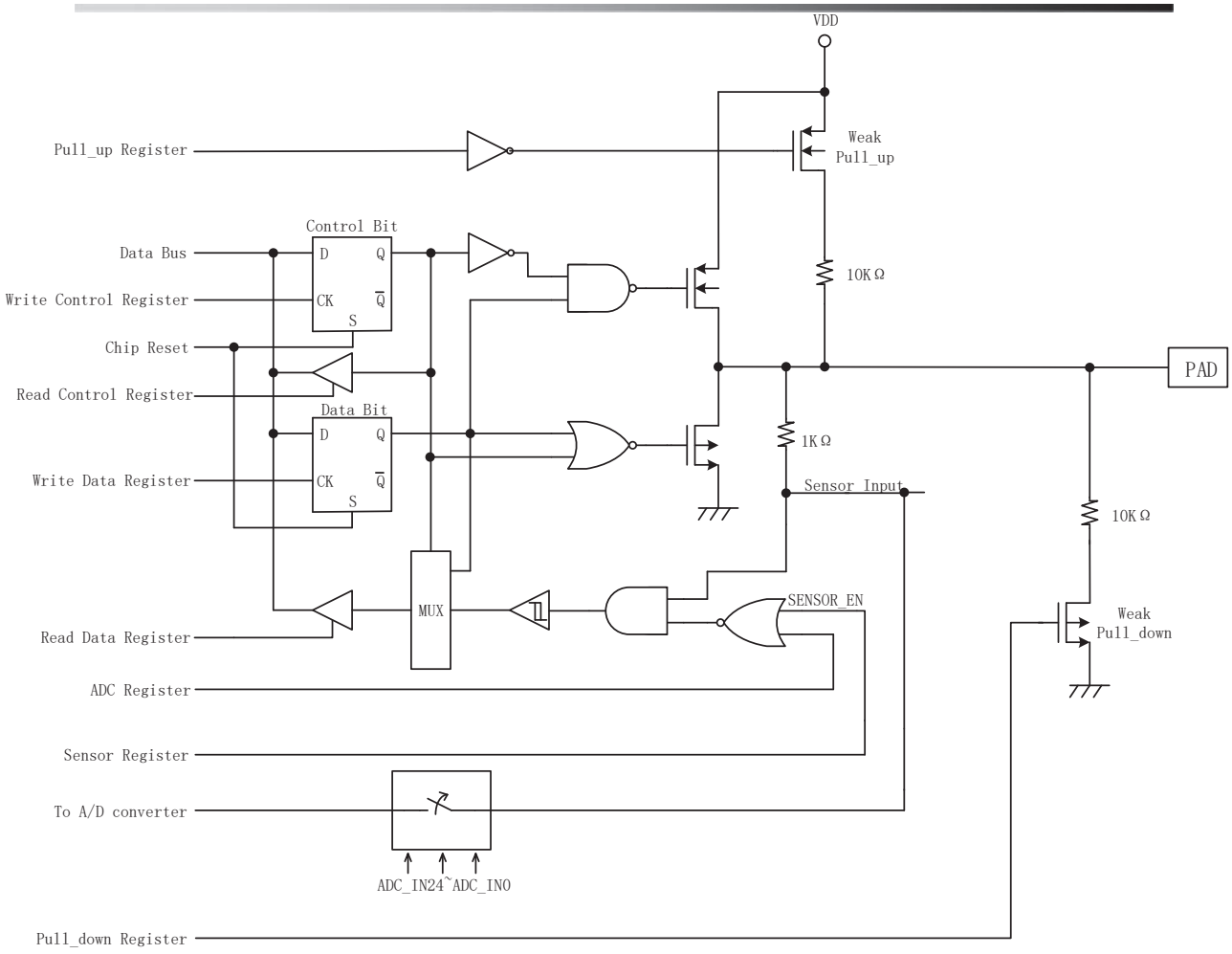


Figure 0.1 Schematic diagram of ADC module structure

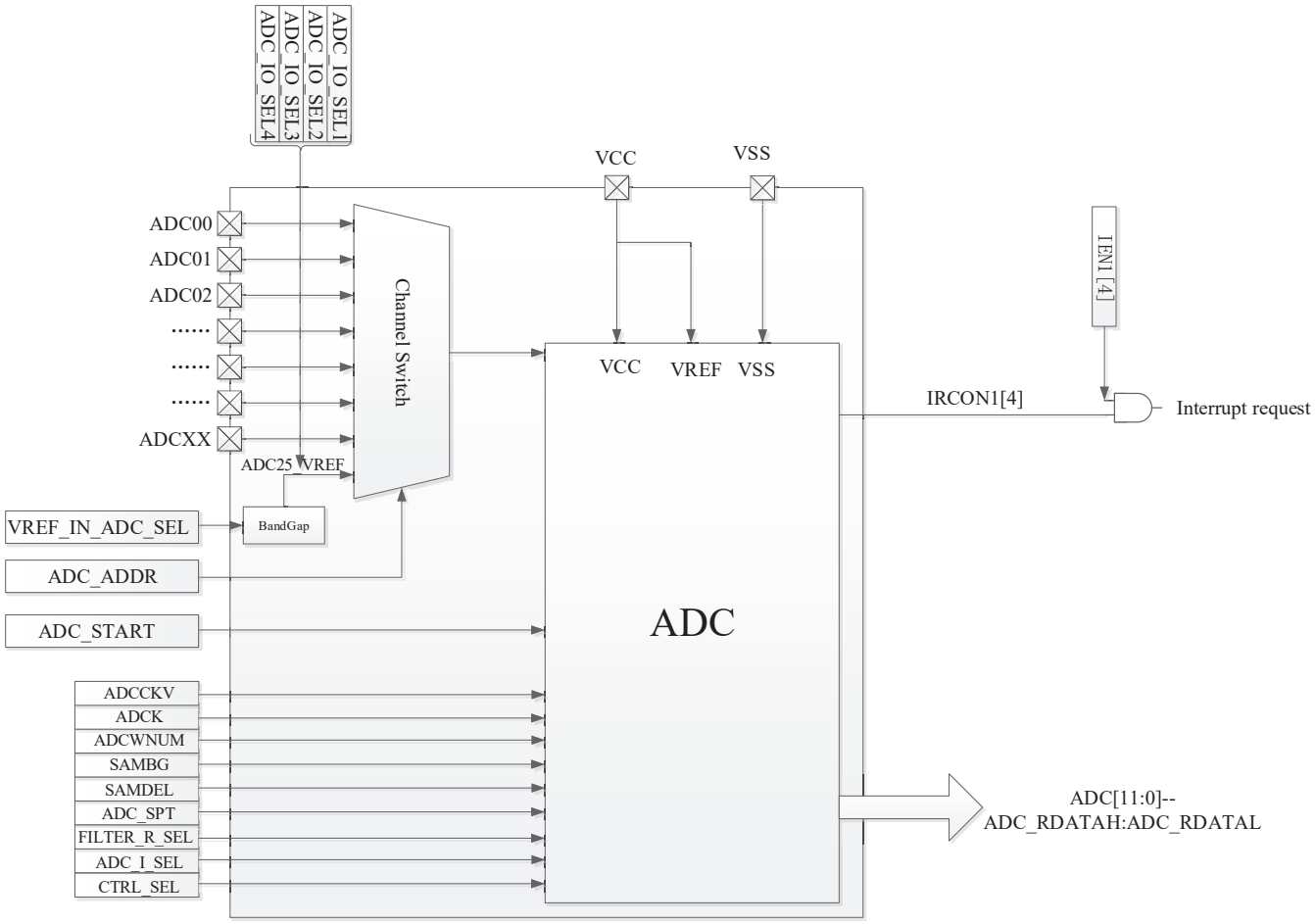


Figure 0.2 ADC Architecture Block Diagram

Channel Selection and Pin Control

The analog inputs have a total of 25 ADC channels.

The pin control register (ADC_IO_SEL) is used to enable/disable the ADC control function of the analog input pins.

The ADC channel address selection register (ADC_ADDR) is used to control the selection of the current scan channel.

ADC_ADDR	Channel
0~24	IO input channel
25	Vref input channel

Table 0.3 Channel Selection

ADC Operating Clock Frequency Division Control

The ADC clock source is PLL_48MHz, and the frequency division clock selection is specified by the ADCK[1:0] bits, as follows:

ADCK	adc_clk Frequency
0	3MHZ
1	2MHZ
2	1.5MHZ
3	1MHZ

Table 0.4 ADC Operating Clock Frequency Division

9.2 ADC Registers

9.2.1 Module Switch Control Register (PD_ANA)

XRAM_SFR Address: 0x2024

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	PD_ADC
Read/Write	-	-	-	-	-	-	-	Read/Write
Po Initial Value	-	-	-	-	-	-	-	1

Bit No.	Symbol	Description
7~1	--	Reserved
0	PD_ADC	Analog ADC Operating Control Register PD_ADC=0 ADC module works normally PD_ADC=1 ADC module does not work

9.2.2 ADC Scan Control Register (ADC_CH_CFG)

Address: 0x97

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	ADC_TRIG_EN	ADC_ADDR					ADC_START
Read/Write	-	Read/Write	Read/Write					Read/Write

Po Initial Value	-	0	0	0

Bit No.	Symbol	Description
6	ADC_TRIG_EN	ADC Trigger Type Selection 1: Hardware Trigger (PWM0_TRGO) 0: Software Trigger
5~1	ADC_ADDR	ADC channel address selection register. 00000: corresponding to ADC0 00001: corresponding to ADC1 …… 11000: corresponding to ADC24 11001: corresponding to ADC25_VREF All other values reserved.
0	ADC_START	ADC scan to start register. 0: ADC module does not scan. 1: ADC module starts scanning ADC_START set from 0 to 1, ADC start scanning, after scanning once, ADC_START hardware automatically set to 0, the corresponding ADC interrupt flag sets to 1, ADC interrupt flag bit needs to clear to 0 by software. **Note**: ADC_START is not allowed to be configured during scanning. **Note:** ADC_START writes 1 to start the scan.After the scan ,the hardware will be set to 0 automatically.To start the next scan,the software will be set to 1 again.If ADC_START = 0is set during the scan,the scan will stop immediately, the module internal related signal is reset. Operation must follow the configuration: ADC_START = 1, interrupt detected, configure ADC_START = 0. ADC_START is not allowed to be configured during the scan.

9.2.3 ADC Scan Result Register Low 8 bits (ADC_RDATAL)

Address: 0x99

Bit No.	7	6	5	4	3	2	1	0
Symbol	ADC_RAWDATAL<7:0>							
Read/Write	Read							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	ADC_RAWDATAL[7:0]	ADC Scan Result Register Low 8 bits

9.2.4 ADC Scan Result Register High 4 bits (ADC_RDATAH)

Address: 0x9A

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	ADC_RAWDATAH[11:8]			
Read/Write	-	-	-	-	Read			
Po Initial Value	-	-	-	-	0			

Bit No.	Symbol	Description
3~0	ADC_RAWDATAH[11:8]	ADC Scan Result Register High 4 bits

9.2.5 ADC Sampling Time Configuration Register (ADC_SPT)

XRAM_SFR Address: 0x2053

Bit No.	7	6	5	4	3	2	1	0
Symbol	ADC_SPT							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	ADC_SPT	ADC sampling time configuration register (0~255) sampling time: sample_Timer = (ADC_SPT+1)*4*Tadc_clk

9.2.6 ADC Sampling Timing Control Register 2 (ADC_ANA_CFG2)

XRAM_SFR Address: 0x2054

Bit No.	7	6	5	4
Symbol	-	-	-	ADCVREF_SELVCC_SEL
Read/Write	-	-	-	Read/Write
Po Initial Value	-	-	-	0
Bit No.	3	2	1	0
Symbol	ADCVREF_SEL	VREF_IN_ADC_SEL		CTRL_SEL

Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	1

Bit No.	Symbol	Description
4	ADCVREF_SELVCC_SEL	0: Output VCC 1: Output 2.048V or 4.096V reference voltage
3	ADCVREF_SEL	0: ADCVREF output 2.048V 1: ADCVREF output 4.096V
2~1	VREF_IN_ADC_SEL	Input to ADC25 internal channel reference voltage selection 00: 1.375; 01: 2.200; 10: 3.143; 11: 4.125 The ADC25 internal channel input voltage calibration value needs to be read during use
0	CTRL_SEL	ADC comparator offset erase selected signal 0: Offset erased after the sampling 1: All switches are disconnected together

9.2.7 ADC Sampling Timing Control Register 1 (ADC_ANA_CFG1)

XRAM_SFR Address: 0x2055

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	ADCWNUM					ADC_I_SEL	
Read/Write	-	Read/Write					Read/Write	
Po Initial Value	0	1	1	1	1	1	1	1

Bit No.	Symbol	Description
6~2	ADCWNUM	Distance conversion interval time selection after sampling is completed 3+ADCWNUM(ADC_CLK)
1~0	ADC_I_SEL	ADC bias current size selection register ADC_I_SEL<0> : comparator bias current size selection signal, 0 for 1uA, 1 for 2uA. ADC_I_SEL<1> : Op-amp bias current size selection signal, 0 is 1uA, 1 is 2uA.

9.2.8 ADC Clock Control Register (ADCCKC)

XRAM_SFR Address: 0x2056

Bit No.	7	6	5	4	3	2	1	0
Symbol	FILTER_R_SEL	SAMBG	SAMDEL		-		ADCK	
Read/Write	Read/Write	Read/Write	Read/Write		-		Read/Write	
Po Initial Value	0	0	0		-		0	

Bit No.	Symbol	Description
7	FILTER_R_SEL	Input signal filter selection. 0: No filter. 1: Adds 10K resistance filter. The default value is 0.
6	SAMBG	Sampling timing and compare timing interval selection: 0: Interval 0; 1: Interval 1 (ADC_CLK)
5~4	SAMDEL	Sampling delay time selection 0:0; 1:2; 2:4; 3:8 (ADC_CLK)
3~2	--	Reserved
1~0	ADCK	ADC_CLK frequency division selection. 0: 3 MHz. 1: 2 MHz. 2: 1.5 MHz. 3: 1 MHz.

9.2.9 ADC Pin Control register 1 (ADC_IO_SEL1)

XRAM_SFR Address: 0x2040

Bit No.	7	6	5	4	3	2	1	0
Symbol	ADC_IO_SEL1[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	SEL_ADC[7:0]	ADC Function Selection 1: Select the ADC function. 0: ADC function is not selected. 00000001=ADC00; 00000010=ADC01. 00000100=ADC02; 00001000=ADC03. 00010000=ADC04; 00100000=ADC05. 01000000=ADC06; 10000000=ADC07

When the ADC IO is selected, it is automatically set as input.

9.2.10 ADC Pin Control register 2 (ADC_IO_SEL2)

XRAM_SFR Address: 0x2041

Bit No.	7	6	5	4	3	2	1	0
Symbol	ADC_IO_SEL2[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	SEL_ADC[15:8]	Disable ADC control of analog input pins. 1: ADC function selected. 0: ADC function not selected. 00000001=ADC08; 00000010=ADC09. 00000100=ADC10; 00001000=ADC11. 00010000=ADC12; 00100000=ADC13. 01000000=ADC14; 10000000=ADC15

When the ADC IO is selected, it is automatically set as input.

9.2.11 ADC Pin Control register 3 (ADC_IO_SEL3)

XRAM_SFR Address: 0x2042

Bit No.	7	6	5	4	3	2	1	0
Symbol	ADC_IO_SEL3[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	SEL_ADC[23:16]	Disable ADC control of analog input pins. 1: ADC function selected. 0: ADC function not selected. 00000001=ADC16; 00000010=ADC17. 00000100=ADC18; 00001000=ADC19. 00010000=ADC20; 00100000=ADC21. 01000000=ADC22; 10000000=ADC23

When the ADC IO is selected, it is automatically set as input.

9.2.12 ADC Pin Control Register 4 (ADC_IO_SEL4)

XRAM_SFR Address: 0x2043

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	ADC_IO_SEL4
Read/Write	-	-	-	-	-	-	-	Read/Write
Po Initial Value	-	-	-	-	-	-	-	0

Bit No.	Symbol	Description
0	SEL_ADC[24]	Disable ADC control of analog input pins 1: ADC function selected. 0: ADC function not selected. 01=ADC24

When the ADC IO is selected, it is automatically set as input.

9.3 ADC Configuration Procedures

Configuration must follow the procedures: ADC_START = 1, interrupt detected, configure ADC_START = 0. ADC_START is not allowed to be configured during the scan.

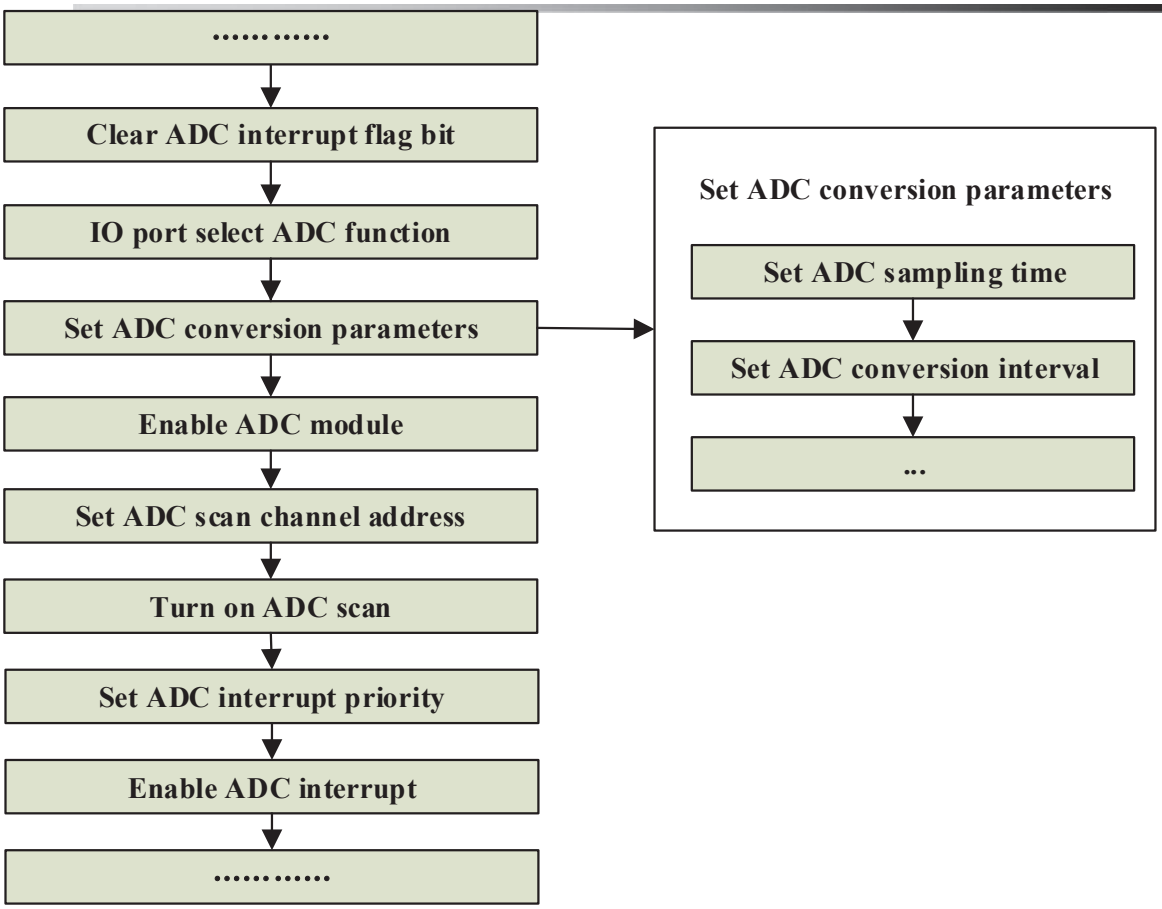


Figure 12.3.1 ADC Configuration Flow Diagram

Table 12.3.2 gives ADC detection time as follows:

Formula	Description
$T_{AD} = T_{ADC\_SPT} + T_{W1} + T_{W2}$	ADC conversion time
$T_{ADC\_SPT}(\mu s) = 4*(ADC\_SPT+1) * T_{adc\_clk}$	ADC sampling time
$T_{W1} = (ADCWNUM+3+SAMDEL) * T_{adc\_clk}$	The period interval between sampling completion and conversion time.
$T_{W2} = 13 * T_{adc\_clk} + 5 * T_{clk\_12M}$	Fixed time

Table 12.3.2 ADC Detection Time Expression

Note:

1) Sequence timing requirements: (ADCWNUM+3)* Tadc_clk >4*Tadc_clk.
   - Voltage setup time after ADC incoming signal adds RC filter >= 2*(ADC sampling conversion time).
   - ADCWNUM: The period interval between sampling completion and conversion time (ADCWNUM =2~31).
   - ADCK: ADC_CLK frequency division selection = 0: 3MHz; 1: 2MHz; 2: 1.5MHz; 3: 1MHz.

- ADC_SPT: ADC sampling time configuration register (ADC_SPT=0~255).
- ADCCKV: ADC analog input clock for comparator offset erase.
- SAMDEL: Sampling delay time selection (SAMDEL = 0:0; 1:2; 2:4; 3:8).

2) ADC conversion time: T= sample_time+ time2+ time3+ time4
   - sample_time= (ADC_SPT+1)*4*Tadc_clk
   - time2=(ADCWNUM+3+SAMDEL)* Tadc_clk(SAMDEL=0:0;1:2;2:4;3:8)
   - time3=13*Tadc_clk
   - time4=5*Tclk_12M
3) When the supply voltage fluctuates or drops, the VCC can be back-calculated by ADCINNER_Data/VREF_IN_ADC_SEL = 4096/VCC, and the Vin can be back-calculated by the Vin_Data/Vin = 4096/VCC.
   - ADCINNER_Data: ADC internal channel data.
   - Vin_Data: ADC input channel data.
   - Vin: ADC input channel voltage.
   - VREF_IN_ADC_SEL: The chip calibration value needs to be read.
   - Vin = (Vin_Data/ADCINNER_Data)*VREF_IN_ADC_SEL, VREF_IN_ADC_SEL needs to read the chip calibration value, first obtain the internal channel data, then obtain the input voltage data Vin_Data. The interval between these two times to obtain data should be as short as possible.
4) Condition for ADC entering into the interrupt: the configuration sequence is ADC_IO_SEL enable -> ADC interrupt enable -> ADC_ADDR (address must correspond to ADC_IO_SEL) -> ADC_START, and note the initialization configuration timing for use by the applications. For application of ADC and IO function multiplexing, it is necessary to notice the switching timing, if ADC_IO_SEL enable is off or the address does not correspond to ADC_IO_SEL can not turn on ADC scan, it must follow the configuration order: ADC_IO_SEL enable -> ADC interrupt enable -> ADC_ADDR (address must correspond to ADC_IO_SEL) -> ADC_START in order to turn on ADC scan.
5) When the pin is configured as ADC function, the pin needs to be configured for IO input mode and other multiplexing functions are off, such as pull-up resistors, etc.

10 TouchKey

10.1 Introduction

BS9000AMxx supports 25 Sensor channels and CSD in parallel, and scans up to 25 channels at a time, which can wake up the system with any key. Any channel can be flexibly configured with registers, including detection rate, detection accuracy, pull-up current value. Detection is carried out in a point scan mode, i.e. the software only gives one scan channel address and its corresponding pull-up current value configuration at a time, and transmits an interrupt after the scan is completed.

CSD features：

- With three modes of CSD charge and discharge clock available
  - Fixed frequency division of the system clock from 6M to 369k
  - PRS 1.5M normal distribution
  - PRS 1.5M Uniform distribution
- CSD counting clock with 24M, 12M, 6M, 4M selectable
- Counting bit width available from 9 bits to16 bits
- Support asynchronous scan mode only
- Support Wake-up in Wait mode
- Supports 25 Sensor channels

The BS9000AMxx implements a wide range of functional applications through a series of registers. The relationship between related values of capacitance detection and SFR values is as follows.

- The size of the count value is proportional to RESO, Rb resistor, PULL_I_SELA_H and inversely proportional to VTH_SEL. It is proportional to the charging and discharging frequency set in PRS_DIV when fully charging and discharging are ensured.
- The variations in channel touch is proportional to RESO and Rb and inversely proportional to VTH_SEL. When fully charging and discharging are ensured, it is proportional to the charging and discharging frequency set through PRS_DIV and the variations in touch.
- The signal-to-noise ratio of touch detection is proportional to VTH_SEL, PULL_I_SELA_L, and inversely proportional to CSD_DS. In case of incomplete charging and discharging, it is inversely proportional to the frequency of charging and discharging set in PRS_DIV and the signal-to-noise ratio.
- The time for individual key detection is related to RESO, CSD_DS.

Note:

The parameters should be configured in such a way that the keys are fully charged and discharged.

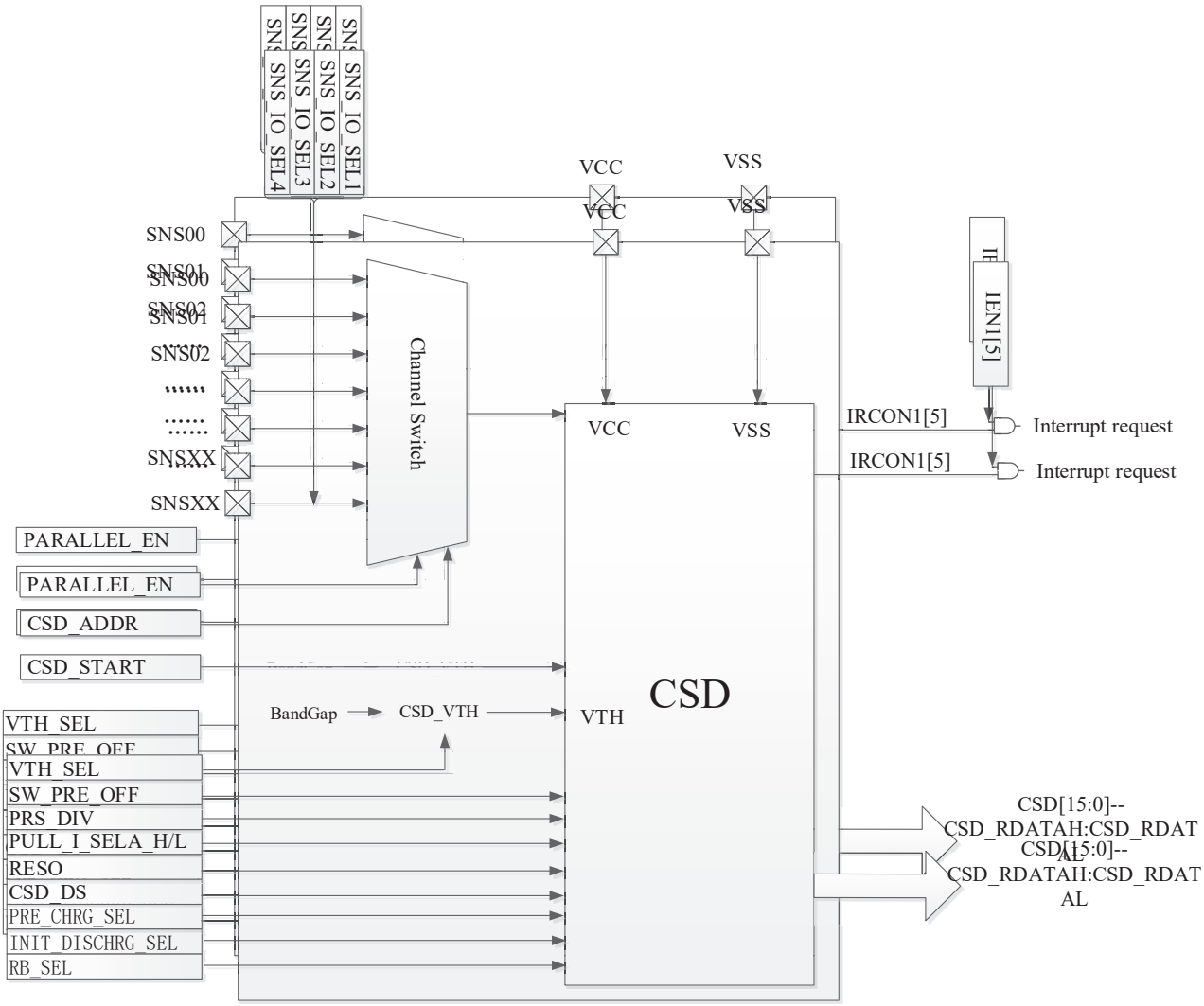


Figure 0.1 CSD Structure Block Diagram

10.2 TouchKey Register

10.2.1 Module Switch Control Register (PD_ANA)

XRAM_SFR Address: 0x2024

Bit No.	7	6	5	4		3	2	0
Symbol	-	-	-	-	-	-	PD_CSD	-
Read/Write	-	-	-	-	-	-	Read/Write	-
Po Initial	-	-	-	-	-	-	1	-

Value								

Bit No.	Symbol	Description
7~2	-	Reserved.
1	PD_CSD	Analog CSD control register. PD_CSD=0 CSD module works normally PD_CSD=1 CSD module does not work
0	-	Reserved.

10.2.2 CSD Start Scanning Register (CSD_START)

Address: 0x92

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	-
Read/Write	-	-	-	-	-	-	-	Read/Write
Po Initial Value	-	-	-	-	-	-	-	0

Bit No.	Symbol	Description
0	--	1: CSD scan starts. 0: CSD scan stops. CSD_START writes 1 to start scanning, after scanning once, the hardware automatically sets to 0. If you want to begin the next scan, you need to set 1 again via software; if CSD_START=0 during scanning, the scan will stop immediately and the related signal inside the module will be reset. **Note:** Must be configured for use in accordance with the process: CSD_START=1, interrupt detected, and configure CSD_START=0. CSD_START is not allowed to be configured during scanning

CSD scan is turned on by software control (in register CSD_START), pulls up by software to turn on the scan, the hardware automatically clears after the scan is finished. To begin the next scan requires to pull up the enable again by software.

If CSD_START =0 is set during scanning, this scan stops and the internal related signals in the module are reset, i.e. to stop CSD scanning at any time, only configuring CSD_START = 0 via software is required.

Note: configuration for use has to follow the procedure: CSD_START = 1, interrupt detected, the configure CSD_START = 0. CSD_START is not allowed to be configured during scanning.

10.2.3 Touch Channel Control Register (SNS_CH_CFG)

Address: 0x93

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	PULL_I_SELA_H	PARALLEL_EN	CSD_ADDR				
Read/Write	-	Read/Write	Read/Write	Read/Write				
Po Initial Value	-	1	0	0				

Bit No.	Symbol	Description
6	PULL_I_SELA_H	Configure the highest bit for CSD pull-up current source **Note:**Pull-up current source high bit. Default value: 0x01.
5	PARALLEL_EN	SNS channels in parallel enable register 1: Multi-channel parallel connection 0: Single channel
4~0	CSD_ADDR	Address of the detection channel, corresponding to channel from 0 to 24.

10.2.4 CSD Pull-up Current Source Selection Register (PULL_I_SELA_L)

Address: 0x94

Bit No.	7	6	5	4	3	2	1	0
Symbol	PULL_I_SEL<7:0>							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	PULL_I_SEL<7:0>	Select switch for CSD pull-up current source size. The default value is 0. **Note:**Pull-up current source low 8 bits. Current source size = 255.5-0.5*{PULL_I_SELA_H, PULL_I_SELA}, the smaller the current source, the smaller the count value. Default value is 0x00.

10.2.5 CSD Count Value Low 8 bits (CSD_RAWDATAL)

Address: 0x95

Bit No.	7	6	5	4	3	2	1	0
Symbol	CSD_RAWDATAL[7:0]							

Read/Write	Read
Po Initial Value	0

Bit No.	Symbol	Description
7~0	CSD_RAWDATAL[7:0]	CSD count value low 8 bits.

10.2.6 CSD Count Value High 8 bits (CSD_RAWDATAH)

Address: 0x96

Bit No.	7	6	5	4	3	2	1	0
Symbol	CSD_RAWDATAH[7:0]							
Read/Write	Read							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	CSD_RAWDATAH[7:0]	CSD count value high 8 bits.

10.2.7 SNS Channel Selection Register 1 (SNS_IO_SEL1)

XRAM_SFR Address: 0x203C

Bit No.	7	6	5	4	3	2	1	0
Symbol	SNS_IO_SEL1 [7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	SNS_IO_SEL1[7:0]	SENSOR IO selection enable bit. 1: SENSOR selected. 0: SENSOR not selected. 00000001=SNS00; 00000010=SNS01. 00000100=SNS02; 00001000=SNS03. 00010000=SNS04; 00100000=SNS05. 01000000=SNS06; 10000000=SNS07

For SENSOR IO selection, set IO as Input automatically.

10.2.8 SNS Channel Selection Register 2 (SNS_IO_SEL2)

XRAM_SFR Address: 0x203D

Bit No.	7	6	5	4	3	2	1	0
Symbol	SNS_IO_SEL2 [7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	SNS_IO_SEL2[7:0]	SENSOR IO Selection enable bit. 00000001=SNS08; 00000010=SNS09. 00000100=SNS10; 00001000=SNS11. 00010000=SNS12; 00100000=SNS13. 01000000=SNS14; 10000000=SNS15

For SENSOR IO selection, set IO as Input automatically.

10.2.9 SNS Channel Selection Register 3 (SNS_IO_SEL3)

XRAM_SFR Address: 0x203E

Bit No.	7	6	5	4	3	2	1	0
Symbol	SNS_IO_SEL3 [7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	SNS_IO_SEL3 [7:0]	1: SENSOR is selected. 0: SENSOR is not selected. 00000001=SNS16; 00000010=SNS17. 00000100=SNS18; 00001000=SNS19. 00010000=SNS20; 00100000=SNS21. 01000000=SNS22; 10000000=SNS23

For SENSOR IO selection, set IO as Input automatically.

10.2.10 SNS Channel Selection Register 4 (SNS_IO_SEL4)

XRAM_SFR Address: 0x203F

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	SNS_IO_SEL4[0]
Read/Write	-	-	-	-	-	-	-	Read/Write
Po Initial Value	-							0

Bit No.	Symbol	Description
0	SNS_IO_SEL4[0]	SENSOR IO selection enable bit. 1: SENSOR and SNS24 selected. 0: SENSOR not selected.

For SENSOR IO selection, set IO as Input automatically.

The above Sensor IO selection does not affect the Sensor enable for scanning and is only used to control the Sensor enable for IO. Also, when connected in parallel, all selected Sensor IO will transmit scan timing.

10.2.11 TouchKey Scan Configuration Register 1 (SNS_SCAN_CFG1)

XRAM_SFR Address: 0x2050

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	SW_PRE_OFF	PRS_DIV					
Read/Write	-	Read/Write	Read/Write					
Po Initial Value	-	0	0					

Bit No.	Symbol	Description
6	SW_PRE_OFF	Front-end charge/discharge clock switch control. 1: disable sw_clk. 0: enable sw_clk.
5~0	PRS_DIV	Front-end charge/discharge clock frequency selection register. 0~61: for fixed frequency: F=F48m/2/(PRS_DIV+4) (6M~369k) 62: highest frequency of 3M, lowest frequency of 1M, center frequency of 1.5M, normal distribution 63: highest frequency of 3M, lowest frequency of 1M, center frequency of 1.5M, uniform distribution

10.2.12 TouchKey Scan Configuration Register 2 (SNS_SCAN_CFG2)

XRAM_SFR Address: 0x2051

Bit No.	7	6	5	4
Symbol	-	RESO		
Read/Write	-	Read/Write		
Po Initial Value	-	1	1	1
Bit No.	3	2	1	0
Symbol	CSD_DS		PRE_CHRG_SEL	NIT_DISCHRG_SEL
Read/Write	Read/Write		Read/Write	Read/Write
Po Initial Value	0		0	0

Bit No.	Symbol	Description
6~4	RESO	Counter bit selection register. 000 : 9 bits; 001 : 10 bits. 010 : 11 bits; 011 : 12 bits. 100 : 13 bits; 101 : 14 bits. 110 : 15 bits; 111 : 16 bits. **Note**：0~7 TouchKey capacitance scan resolution, and the number of counter bits equals to (RESO + 9) bits**.** The larger the TouchKey capacitance scan resolution, the larger the drop in Raw Data variations, while the noise introduced will also increase, otherwise the opposite.
3~2	CSD_DS	Count clock frequency selection register. 00:24M; 01:12M. 10: 6M; 11: 4M; Default as 0 **Note**：Detection speed **0: 24M, 1: 12M, 2: 6M, 3: 4M.** The smaller the detection speed, the slower the time to sample Raw data, otherwise the opposite. It is recommended that setting the fastest 24M as the default, and the detection speed is at least 2 times the PRS clock.
1	PRE_CHRG_SEL	Precharge time selection. 0: 20us 1: 40us
0	NIT_DISCHRG_SEL	Pre-discharge time selection. 0: 2us 1: 10us

10.2.13 TouchKey Scan Configuration Register (SNS_ANA_CFG)

XRAM_SFR Address: 0x2052

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	RB_SEL			VTH_SEL		
Read/Write	-	-	Read/Write			Read/Write		
Po Initial Value	-	-	1	0	1	1	0	1

Bit No.	Symbol	Description
5~3	RB_SEL	Rb resistor size selection 0: 20k; 1: 40k. 2: 60k; 3: 80k; 4-7: 100k When using, you need to read the Rb80k calibration value from the chip FLASH: CBYTE[0x41CD]k/80k, to calculate the normalized sensitivity proportionally. **Note:** Rb resistance selection: 010: 60k, 011: 80k, others: 100k. The larger the resistance, the greater the variations in Raw data, while the noise introduced will also increase, otherwise the opposite.
2~0	VTH_SEL	VTH voltage selection signal 000: 1.505V, 001: 2.133V, 010: 2.508V. 011: 2.859V, 100: 3.177V, 101: 3.487V. 110: 3.864V, 111: 4.205V **Note:** 1~7, the smaller the reference voltage, the greater the variations in Raw Data, while the noise introduced will also increase, otherwise the opposite.

10.3 TouchKey Configuration Process

The TouchKey scan is query or interrupt mode, first, configure the TouchKey parameters; then, turn on the TouchKey scan; finally, obtain and save the TouchKey data at the TouchKey interrupt, and the software algorithm carries out the manipulation of the data and determines touchkeys output.

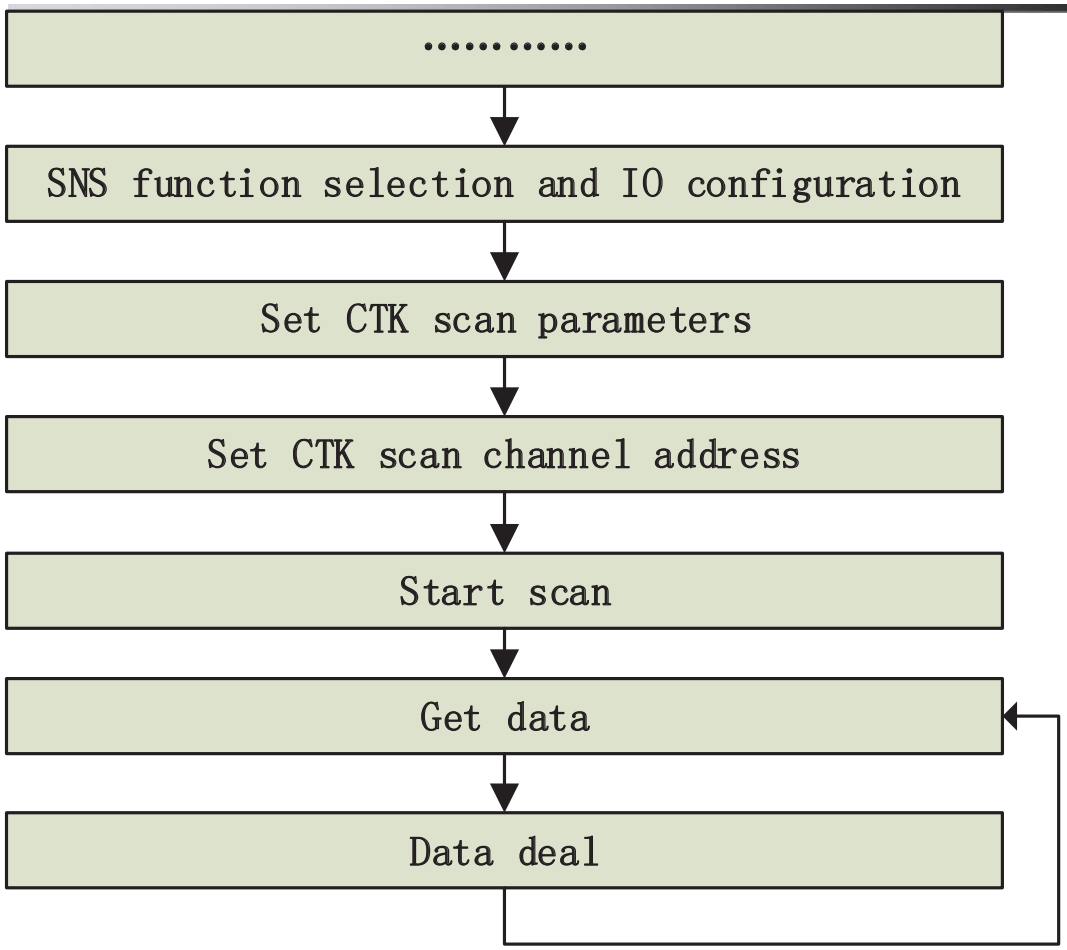


Figure 0.2 TouchKey Scan Configuration Process

The sensitivity parameters are configured to obtain a set of parameters with a better signal-to-noise ratio, thus improving the accuracy of keys pressing judgments.

Note:

1) **Raw data is the real-time raw count value of the capacitive TouchKey counter.**
2) **The application requires to check the data via a debugging software and compare the parameters to get a set of parameters with good signal-to-noise ratio.**
3) **The relation between chip supply voltage and reference voltage: VCC - VTH > 0.5V.**

11 IIC

IIC is for the exchange of data between chips. The host turns on the communication and decides the direction of data transfer, and the operating rate is also determined by the host. Its functions and features are as follows.

- Two serial IOs: serial data line (SDA) and serial clock line (SCL)
- Compliant with PHILIPS Standard Communication Protocol
- Data Transfer Rate: 100Kbps, 400Kbps
- IIC Slave supporting 7-bit addressing
- With clock low level stretching
- Wake up the core via IIC interrupt in low power mode
- Detecting write conflicts and BUF overflow exceptions
- With the function of caching host write data and preloading the host to read data
- Interrupt wake-up mechanism to wake up the CPU when the CPU is configured to enter wait mode.

The IIC master-slave connection diagram is shown below.

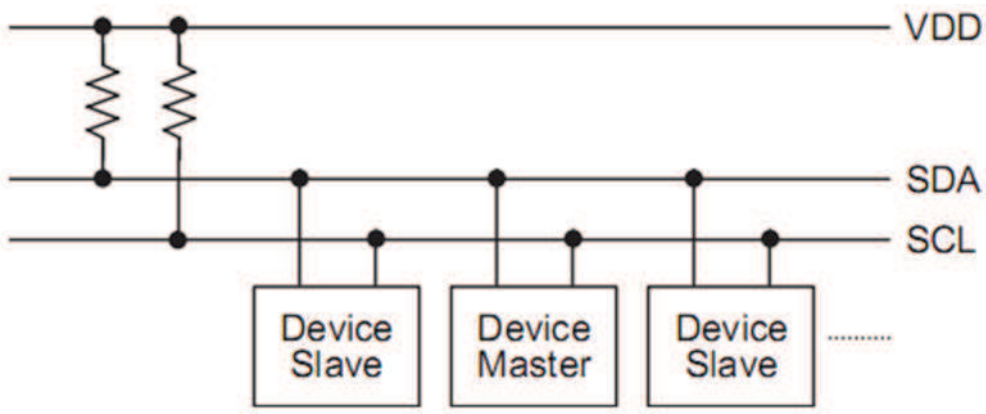


Figure 0.1 IIC Master/Slave Connection

Supports 7-bit Addressing

Only 7-bit addressing is supported, it is not compatible with 10-bit addressing. The slave address is configured by CPU.

Address is equivalent to an ID for the device. According to the IIC protocol, the host sends the slave address code after the S signal, and the slave compares it with the contents of the slave address after receiving the address code to verify its identity. Only when the addresses match can communication take place.

Note: After correctly addressing the slave address, a read/write data operation must follow, otherwise the communication will be affected.

Data Transfer Rate

Fast communication 400Kbps in IIC communication mode and standard communication rate 100Kbps are supported. The communication rate ranges from 50Kbps to 500kbps.

Exception Communication

If any exception occurs during communication, the cause can be determined by the bus status. In the case of write conflict and BUF overflow exceptions, it can be judged by querying the value of register IICSTAT.

Stretch clock low level

IIC is based on the interrupt service subroutine to complete the communication. Each byte received by IIC will generate an interrupt signal to inform the MCU to process the data accordingly, so that the data can be transmitted or received. When the interrupt service subroutine takes too long to process the data normally, it can be configured through the registers to pull down the clock line and stretch the low time to restore the normal communication. For the specific way to stretch the clock low level, refer to the register IICCON description and IIC sequence timing description.

Wakeup from interrupt mechanism

When the CPU is configured to enter sleep mode, it starts IIC communication and after sending an address to match with the slave address, the slave responds with an ACK to generate an IIC interrupt, which will be used to wake up the system.

Note: When configuring the slave to pull down clock line, it wakes up in IDEL mode, the internal SCLEN signal pull-down clock line function is at the falling edge of the ninth clock; if the host supports the pull-down clock line function or the host waits 1ms after the ninth clock falling to send next IIC rising edge, the first byte of the wakeup communication is normal.

11.1 Communication Sequence

The BS9000AMxx uses a hardware slave. When the host Reads and Writes data, the slave receives the address and then generates an interrupt to send a valid acknowledge signal if that address matches. The interrupt will be generated after the eighth clock of the host written data, however the interrupt signal will not be generated when the host sends a stop signal. The following four read and write sequences shows the conditions with F_sys_clk clock. In sleep mode, the IIC_START signal maintains low until the start signal of the next communication arrives. See IIC communication sequence diagram below.

11.1.1 IIC Host Write Sequence

The IIC host write sequence is shown in the following diagram.

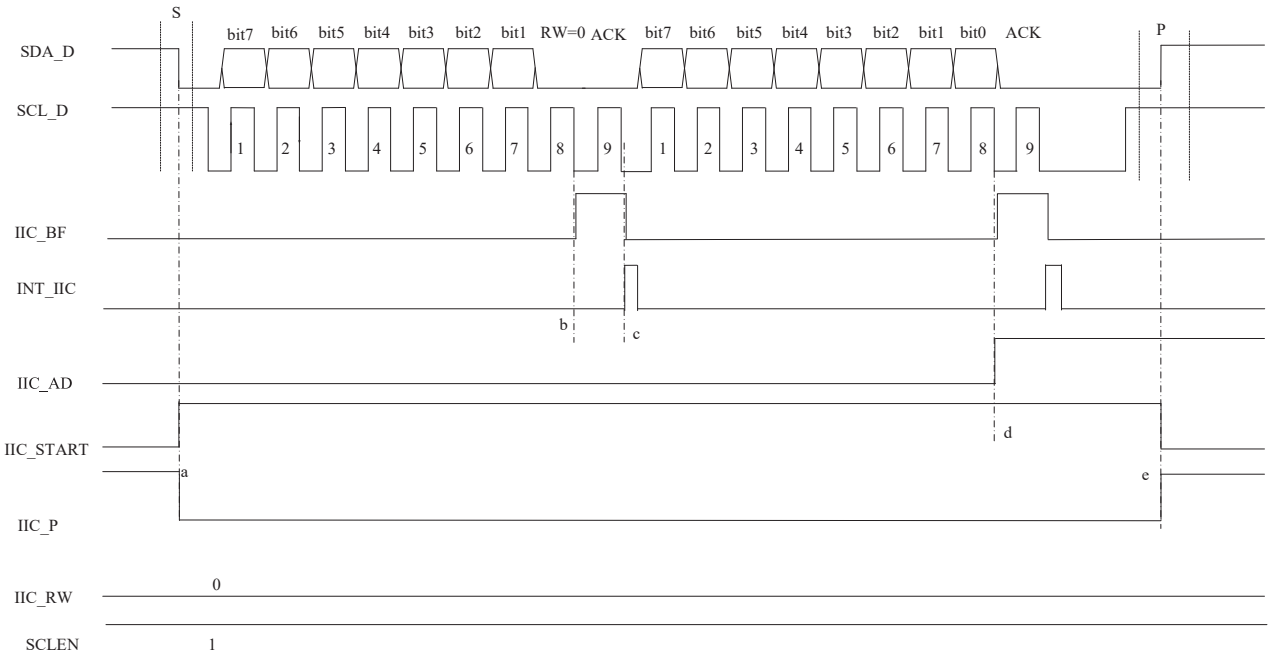


Table 11 .2 IIC write without pulling down the clock line

As shown above is a schematic of the host write operation without pulling down the clock line, from which you can see how the IIC bus changes and how some internal signals change.

The IIC bus changes in accordance with the protocol requirements of the IIC, and the start signal as well as the transmission of the message are in accordance with the protocol requirements. First the host sends the start signal IIC_START, and the slave detects the IIC_START signal and then sets the IIC_START status bit, as shown by the dashed line a in Figure 11.2.

Then the host sends the address byte and the read/write flag bit, the slave automatically compares with its own address after receiving the address byte, and if it matches, it sets IIC_BF after the falling edge of the 8th clock, as shown in dashed line b.

The interrupt signal INT_IIC is generated after the falling edge of the 9th clock, as shown in dashed line c. The MCU needs to read the IICBUF during the execution of the interrupt subroutine, even if this data is useless; the operation of reading the IICBUF causes an indirect clear of STAT_BF.

Then the host continues to send the message, after the falling edge of the 8th clock of the 2nd byte IIC_BF is also set, and the IIC_AD flag bit will also be set, marking the current received byte is data, as shown in the dashed line d. The stop signal has no effect on the IIC_AD flag bit, that is, the stop signal IIC_STOP is detected, the IIC_AD flag bit will not be cleared; The same interrupt will be generated after the falling edge of the 9th clock, and the interrupt subroutine needs to do the same operation.

If the host wants to send more than one byte, it can continue to send. The above diagram only

illustrates the case where the host sends one data. Finally, the host sends a stop signal IIC_STOP after sending all the data, which marks the end of communication and releases the IIC bus to the idle state.

11.1.2 IIC Host Write Pull-down Sequence

The IIC host write pull-down sequence are shown in the following diagram.

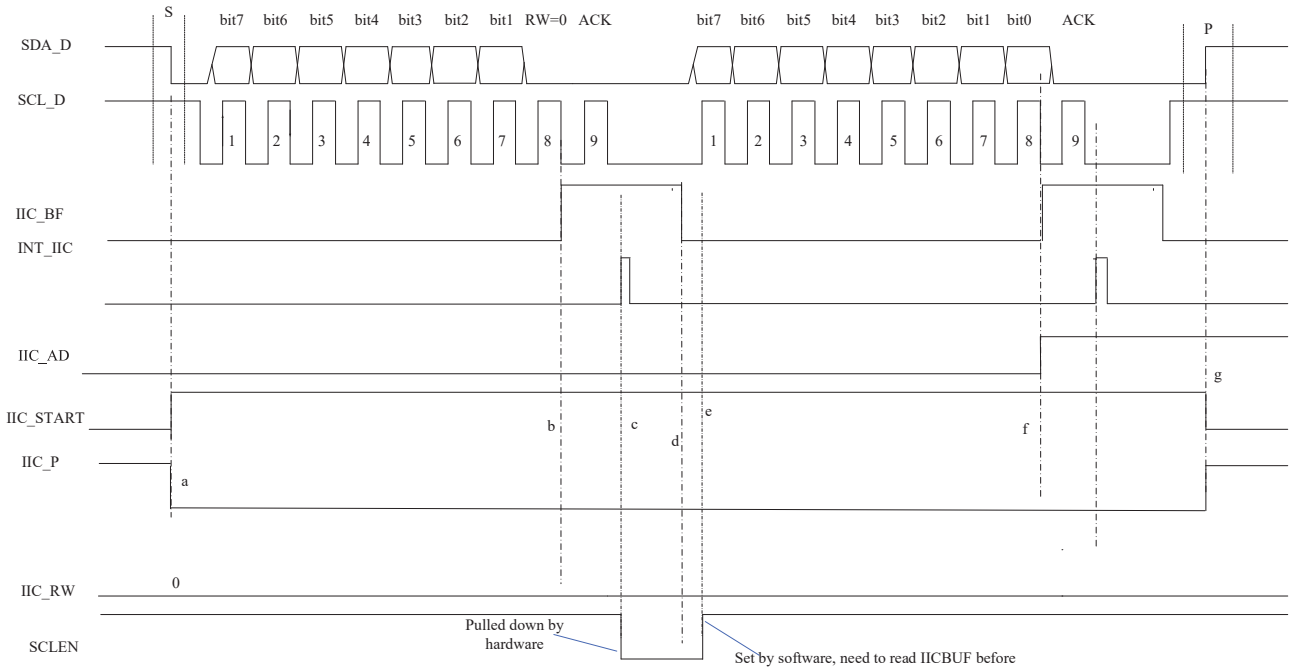


Table 11 .3 Host write pull-down clock line diagram

As shown above is a schematic of pulling down the clock line during a host write operation, from which you can see how the IIC bus changes and how some internal signals change.

The IIC bus changes in accordance with the protocol requirements of the IIC, and the start signal as well as the transmission of the telegrams are in accordance with the protocol requirements. First the host sends the start signal IIC_START, and the slave detects the IIC_START signal and then sets the IIC_START status bit, as shown by the dashed line a in Figure 11.2.

Then the host sends the address byte and the read/write flag bit, the slave automatically compares with its own address after receiving the address byte, and if it matches, it sets IIC_BF after the falling edge of the 8th clock, as shown in dashed line b. After the falling edge of the 9th clock, the interrupt signal INT_IIC is generated, as shown in the dashed line c.

After the falling edge of the 9th clock SCLEN is automatically cleared by hardware, this period is used for slave processing or reading data, even if this data is not used, the operation of reading IICBUF will cause the IIC_BF to be cleared indirectly, as shown in dashed line d. Then the software sets SCLEN to release the clock line, as shown in dashed line e.

Then the host will continue to send synchronous clock after detecting the release of SCL from the

slave. After the falling edge of the 8th clock of the 2nd byte, IIC_BF will be set as well, and the IIC_AD flag bit will be set at the same time, marking the current received byte as data, as shown in the dashed line f. The stop signal has no effect on the IIC_AD flag bit that the stop signal IIC_STOP is detected. IIC_AD flag bit will not be cleared; the same interrupt will be generated after the falling edge of the 9th clock.

If the host wants to send more than one byte, it can continue to send, and the schematic illustrates the case where the host sends only one data. Note that when the host sends the last data, it does not enable the pull-down clock line function. Finally, the host sends a stop signal IIC_STOP after sending all the data, marking the end of communication and releasing the IIC bus into the idle state.

11.1.3 IIC Host Read Sequence

The IIC host read sequence is shown in the following diagram.

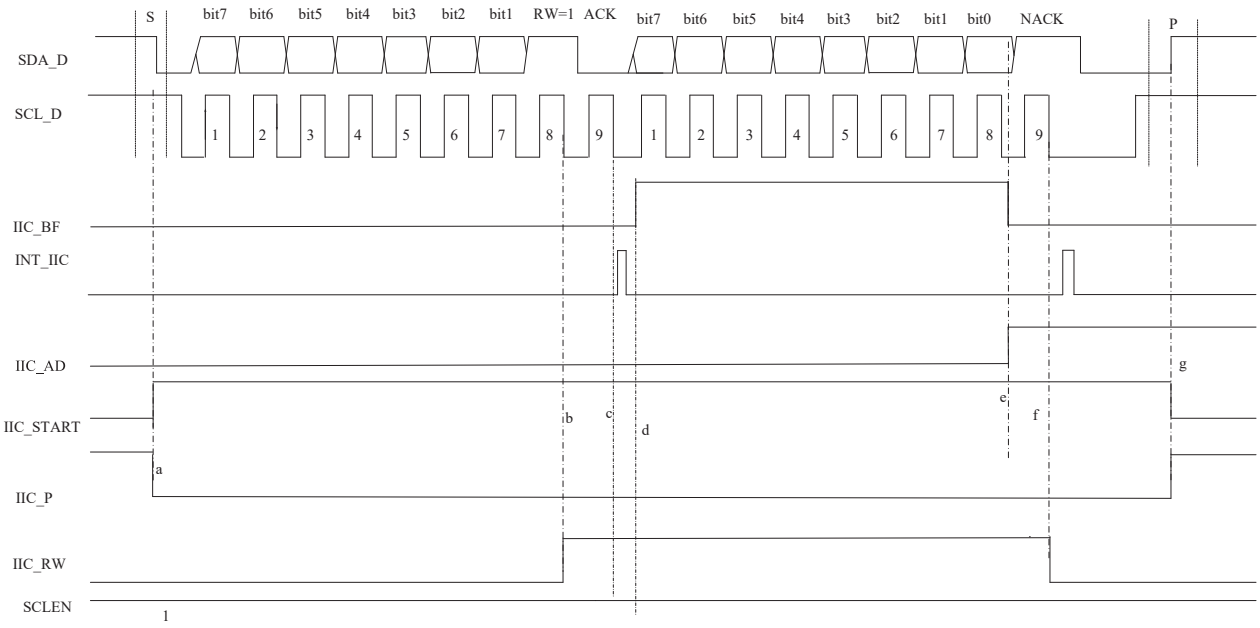


Table 11 .4 IIC Host Read without Pulling Down Low Clock Line Diagram

As shown in the figure above, it is the sequence diagram of the master reading the slave clock line pulled down. From the diagram, you can know how the bus changes, and how some internal signals of the circuit change.

The host sends the IIC_START signal to mark the start of communication, as shown in dashed line a. The internal circuitry detects the IIC_START signal sequence and sets the status flag bit IIC_START.

The host sends the address byte after the IIC_START signal and IIC_RW = 1, indicating that the host reads the slave. The status bit IIC_RW is set after the falling edge of the eighth clock in case of address matches, as shown in dashed line b. IIC_RW is not set if the address does not match.

An interrupt signal is generated after the falling edge of the ninth clock, as shown in dashed line c. And the data in IICBUFFER is ballasted into IICBUF, IIC_BF is set, as shown in dashed line d, and

the highest bit is sent to the bus. After 8 clock's, a byte of data is sent and the IIC_BF flag bit is cleared; also the address data flag bit is set to indicate the current byte of data sent. As shown in dashed line e.

An interrupt is generated after the falling edge of the 9th clock. If the host needs to continue reading the slave, the host replies with a valid acknowledge bit ACK and continues the communication; if the data the host needs has been read, the host replies with an invalid acknowledge bit NACK and then sends the stop signal IIC_STOP to terminate the communication. In the schematic diagram, the host responds NACK after reading just one data, and then sends the IIC_STOP signal to terminate the communication.

The read/write flag bit IIC_RW is cleared by hardware when NACK is detected, as shown by dashed line f.

If the host sends NACK, the slave SCLEN is not automatically pulled down, which should be noted in the application.

When the IIC_STOP signal is detected, the status bit IIC_STOP is set and IIC_START is cleared to zero, as shown in dashed line g.

11.1.4 IIC Host Read Pull-down Sequence

The IIC host read pull-down sequence is shown in the following diagram.

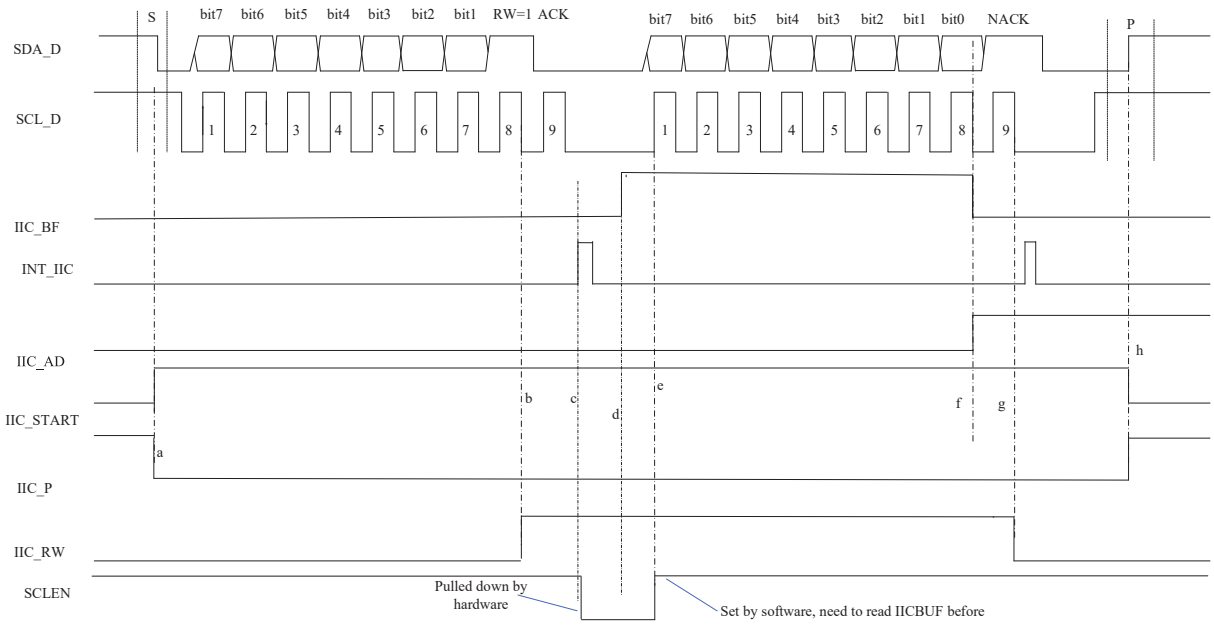


Table 11 .5 IIC host reads out pull-down clock line diagram

As shown in the figure above, it is the sequence diagram of the master reading out the slave clock line pulled down. From the diagram, you can know how the bus changes, and how some internal signals of the circuit change.

The host sends the IIC_START signal to mark the start of communication, as shown in dashed line a. The internal circuitry detects the IIC_START signal timing and sets the status flag bit IIC_START.

The host sends the address byte after the IIC_START signal and IIC_RW = 1, indicating that the host reads the slave. The status bit IIC_RW is set after the falling edge of the eighth clock in case of address matches, as shown in dashed line b. IIC_RW is not set if the address does not match.

An interrupt signal is generated after the falling edge of the ninth clock, as shown in dashed line c. SCLEN is also automatically pulled down by hardware after the falling edge of the 9th clock. This period is used for slave processing or preparing data, and then the prepared data is written to the IICBUF, and then SCLEN is set by software to release the clock line. As shown in dashed line d, after writing data into the IICBUF, IIC_BF is set to mark that this IICBUF is full. As shown in dashed line e, the software sets SCLEN and releases the clock line.

Then the host, after detecting the release of SCL from the slave, will continue to send synchronous clocks to read out the data from the slave. After the falling edge of the 8th clock, a byte of data is sent and the IIC_BF flag bit is cleared; at the same time, the address data flag bit is also set to indicate the current byte of data sent. As shown in dashed line f.

An interrupt is generated after the falling edge of the 9th clock. If the host needs to continue reading the slave, it responds with a valid response bit ACK and continues the communication; if the data the host needs has been read, it responds with an invalid response NACK and then sends the stop signal IIC_STOP to terminate the communication. In the schematic diagram, the host responds NACK after reading just one data, and then sends the IIC_STOP signal, which terminates the communication.

The read/write flag bit IIC_RW is cleared by hardware when NACK is detected, as shown in dashed line g.

When the IIC_STOP signal is detected, the status bit IIC_STOP is set and IIC_START is cleared to zero, as shown in dashed line h.

11.1.5 IIC Host Write Data

The IIC host writes data is shown in the following figure.

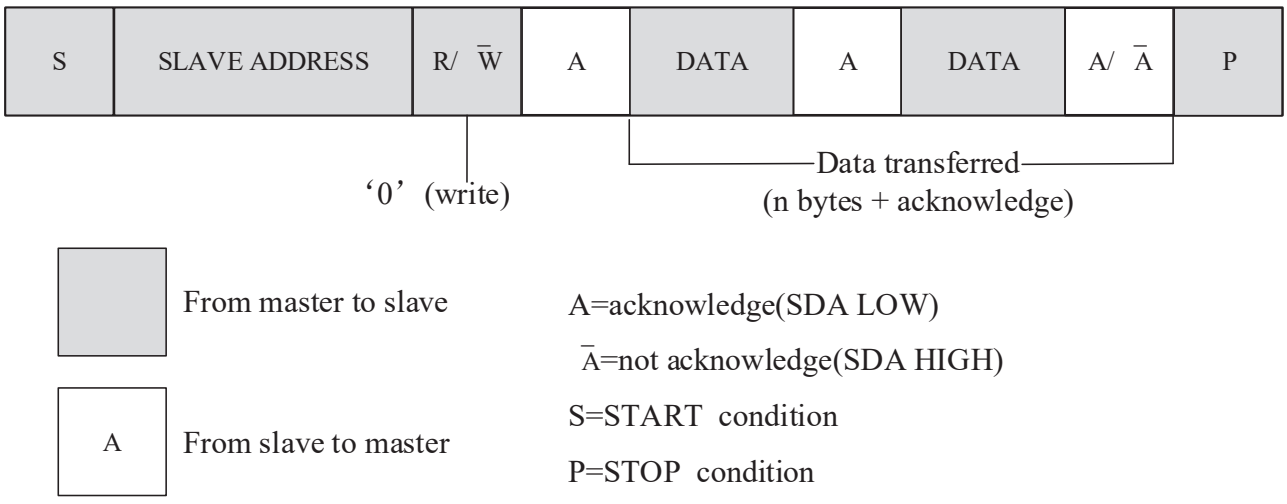


Table 11 .6 IIC host writes data

11.1.6 IIC Host Read Data

The IIC host reads out the data as shown in the following figure.

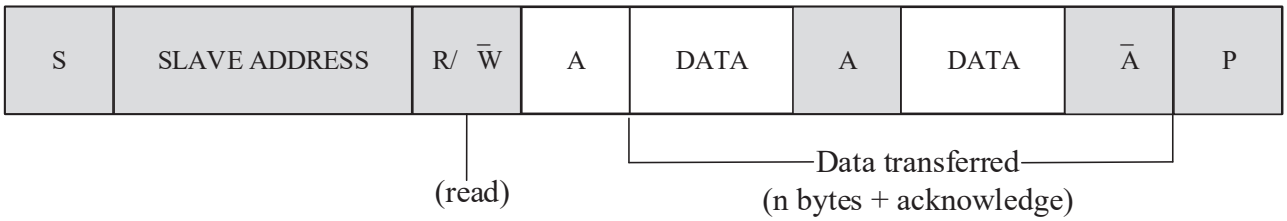


Table 11 .7 IIC host reads out data

The slave gives the ACK signal on the falling edge of the eighth clock, and generates the IIC interrupt after the falling edge of the ninth clock. It is recommended that the host delay 60μs to 300μs after sending the falling edge of the ninth clock to reserve the slave IIC interrupt service data preparation time (if the slave IIC interrupt service processing time is around 100us, it is recommended to delay > 200us), and then send clock signal.

Note: For IIC communication >= 100K, the recommended system clock is 6MHz.

11.2 IIC Registers

11.2.1 IIC Address Register (IICADD)

Address: 0xF7

Bit No.	7	6	5	4	3	2	1	0
Symbol	IICADD[7:1]							-
Read/Write	Read/Write							-
Po Initial Value	0							-

Bit No.	Symbol	Description
7~1	IICADD[7:1]	IIC Address Register

11.2.2 IIC Status Register (IICSTAT)

Address: 0xF8

Bit No.	7	6	5	4
Symbol	IIC_START	IIC_STOP	IIC_RW	IIC_AD
Read/Write	Read	Read	Read	Read
Po Initial Value	0	1	0	0
Bit No.	3	2	1	0
Symbol	IIC_BF	IIC_ACK	IIC_WCOL	IIC_RECOV
Read/Write	Read	Read	Read/Write	Read/Write
Po Initial Value	0	1	0	0

Bit No.	Symbol	Description
7	IIC_START	Start signal status bit. 1 = Indicates that the start bit is detected. 0 = Indicates no start bit detected **Note:** IIC_START will be set when the start signal is detected, indicating that the bus is busy.
6	IIC_STOP	Stop signal flag bit. 1 = Indicates that it is in stop state. 0 = Indicates that no stop bit is detected **Note:**IIC_STOP is set when the device is in stop state, indicating that the bus is idle, and is cleared by hardware when the start signal is detected, indicating the start of communication.
5	IIC_RW	Read/Write flag bit: records the Read/Write information obtained from the address byte after the recent address match 1 = Indicates a read operation. 0 = Indicates a write operation. **Note:**This flag bit records the read and write information bits obtained from the address byte after address matching. IIC_RW=1 means the host reads the slave operation, IIC_RW=0 means the host writes the slave operation. The start signal, stop signal, and non-answer signal (NACK) all clear IIC_RW. The change of this status bit occurs on the falling edge of the 9th clock.
4	IIC_AD	Address data flag bit 1 = Indicates that the most recently received or sent byte is data. 0 = Indicates that the most recently received or sent byte is the address.

		Note: It marks the current received or sent byte is address or data. The start signal, stop signal, and non-valid acknowledge signal have no effect on this status bit. The change of this status bit occurs at the falling edge of the 8th clock.
3	IIC_BF	IICBUF full flag bit. When receiving in IIC bus mode, 1: Indicates that the reception was successful and the buffer is full. 0: Means the reception is not completed and the buffer is still empty When sending in IIC bus mode： 1: Indicates that data transmission is in progress (excluding the answer and stop bits) and the buffer is still full; 0: Indicates that data transmission has been completed (excluding the answer and stop bits) and the buffer is empty. **Note:** It marks that the transceiver buffer is currently full or empty. IIC_BF=0 means the buffer is not receiving data and the buffer is empty; IIC_BF=1 means the buffer is receiving data and the buffer is full. This status bit can only be set and cleared indirectly, and cannot be operated directly. If the address matches and IIC_RW=0, IIC_BF is set after the falling edge of the 8th clock, signaling that the IICBUF received data. The IICBUF should be read out during the execution of the interrupt routine, and the operation of reading the IICBUF will indirectly clear the BF flag bit. If the IICBUF is not read and the host continues to send data, a receive overflow will occur and although the slave still receives the data sent by the host and ballasts it to the IICBUF, it will still send a NACK signal giving an invalid answer. If the address matches and IIC_RW=1, the IIC_BF flag bit will not be set after the slave receives the address byte; IIC_RW=1 means the host reads out the slave operation and the slave needs to write the data to IICBUF, the slave operation of writing IICBUF will set IIC_BF, then the software sets SCLEN and releases the clock line; the host will then send the synchronous clock. After the 8th clock, IIC_BF is cleared by hardware after the data in IICBUF is sent out.
2	IIC_ACK	Answer flag bit 1: Indicates an invalid answer signal. 0: Indicates a valid answer signal. **Note:** Regardless of whether the host is reading or writing, the

		slave will sample the data line at the rising edge of the 9th clock to record the acknowledge information. The acknowledge bit is divided into valid acknowledge bit ACK and non-valid acknowledge bit NACK, that is to say, if the rising edge of the 9th clock samples data "0", it means valid acknowledge ACK, and IIC_ACK is cleared, if it samples data "1", IIC_ACK is set, it means non-valid acknowledge ACK. IIC_ACK is set, indicating non-acknowledge. After the non-acknowledge signal, the host will send a stop signal to announce the end of communication.The start signal will clear this status bit.
1	IIC_WCOL	Write conflict flag bit 1: Indicates that the IIC is sending the current data when new data is trying to write to the transmit buffer; the new data cannot be written to the buffer. 0: No write conflict occurred. **Note:**IICBUF can only be written by CPU if IIC_RW=1, and RD_SCL_EN=1, SCLEN=0. Any other attempt to write the IICBUF is prohibited. If a write operation occurs without meeting the above conditions, the data will not be written to the IICBUF, and the write conflict flag bit IIC_WCOL is set, indicating that a write conflict has occurred, and this flag bit needs to be cleared by software.
0	IIC_RECOV	Receive overflow flag bit 1: Indicates that new data is received while the previous data received by the IIC has not been taken away, and the new data cannot be received by the buffer. 0: Indicates that no receive overflow has occurred. **Note:**If the IICBUF is full, that is, if there is data in the IICBUF, and the IIC has received new data, a receive overflow will occur and IIC_RECOV will be set, while the data in the IICBUF will not be updated and the newly received data will be lost. This status bit also needs to be cleared by software, otherwise it should affect the communication process later. This situation will only occur when IIC_RW=0, BF=1, and the CPU does not read the IICBUF.

11.2.3 IIC Transmit/Receive Data Register (IICBUF)

Address: 0xF9

Bit No.	7	6	5	4	3	2	1	0

Symbol	IICBUF
Read/Write	Read/Write
Po Initial Value	0

Bit No.	Symbol	Description
7~0	IICBUF	IIC transmit/receive data buffer

The specific application process is as follows.
In transmit state, after ballasting the data into the IICBUF, the data is sent out sequentially shifted under the synchronization clock of the host, with the high bit in front. 8 clocks later, a byte is sent out.
In receive state, the data is written to the BUF after 8 clocks have passed in the host, and an interrupt is generated after the 9th clock to tell the CPU to read out the data in the IICBUF.
The operation of writing data to IICBUF is conditional, only when RD_SCL_EN=1 and IIC_RW=1 and SCLEN=0 can write data to IICBUF; otherwise the operation of writing IICBUF is prohibited. In other words, if the condition is not satisfied, the operation of writing IICBUF cannot be successful, the data cannot be written, the data of IICBUF will not change, and it will also cause write conflict.
For example: IICBUF already has data 55h, in the case of write IICBUF conditions are not met but still wanting to write data 0x00 to IICBUF, the result is that the data in IICBUF is still 0x55, while the write conflict flag bit IIC_WCOL is set, informing the user that the operation is abnormal.
When RD_SCL_EN=0, the data to be sent by the slave is the value of the ballasted IICBUFFER register when the interrupt signal is generated.

11.2.4 IIC Status Register (IICCON)

Address: 0xFA

Bit No.	7	6	5	4
Symbol	-	-	IIC_RST	RD_SCL_EN
Read/Write	-	-	Read/Write	Read/Write
Po Initial Value	-	-	0	1
Bit No.	3	2	1	0
Symbol	WR_SCL_EN	SCLEN	SR	IIC_EN
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7~6	--	Reserved bit.
5	IIC_RST	IIC module reset signal 1: IIC module is reset. 0: IIC module works normally

		Note: configuration 1 to reset the IIC module. The IIC_RST is a global reset. Therefore, write the IIC_RST bit to 0 before operating other register.
4	RD_SCL_EN	Host reads pull-down clock line control bit 1: Enables the host read pull-down clock line function. 0: Disable the host read pull-down clock line function. **Note:**RD_SCL_EN is the read pull-down line control bit, which enables the interrupt pull-down clock line function when it is 1, and disable the interrupt pull-down clock line function when it is 0. When RD_SCL_EN=1, SCLEN is automatically pulled down by hardware when the slave receives an address byte or finishes sending a byte and the host sends an ACK, forcing the host into Wait state. To release the IIC clock, the slave needs to do the following two operations: first write the data to be sent to the IICBUF, and then set SCLEN by software. The purpose of this design is to ensure that the data to be sent has been written to the IICBUF before pulling up SCL . When RD_SCL_EN=0, when the slave receives the address byte or finishes sending a byte and the host sends ACK, the slave will immediately ballast the data prepared in the IICBUFFER register to the transmit buffer register and then send it to the data line. Therefore, to ensure the correct data is transmitted each time, the IICBUFFER in the interrupt service routine prepares the next data to be sent, and the data received by the host are the data processed in the last interrupt, and the first received data is prepared in the initialization.
3	WR_SCL_EN	Host write pull-down clock line control bit, 1: Enables the write pull-down clock line function. 0: Disables the write pull-down clock line. **Note:**WR_SCL_EN is the write pull-down line control bit, which enables the interrupt pull-down clock line function when it is 1, and does not enable the interrupt pull-down clock line function when it is 0. With IIC_RW=0, the decision to pull down the clock line can be made based on the host's communication rate and the time to process the interrupt, i.e., configure the WR_SCL_EN bit. When the CPU can handle the interrupt and exit the interrupt within 8 IIC clocks, WR_SCL_EN=0 does not enable the function of pulling down the clock line, then the hardware will not automatically pull down the clock line when the interrupt occurs.

		When the CPU can not handle the interrupt and exit within 8 IIC clocks, WR_SCL_EN=1 enables the function of pulling down the clock line, then the hardware automatically pulls down the clock line when the interrupt occurs, forcing the host to enter Wait state, and the software sets SCLEN when the data written to the IICBUF is read out by the CPU.
2	SCLEN	IIC clock enable bit. 1: The clock is working properly. 0: Pull down the clock line. **Note:**Although the slave cannot generate the communication clock, the slave can stretch the low voltage time of clock according to the protocol. SCLEN=0, clock line is locked at low voltage, SCLEN=1 means release the clock line. Stretching the clock low voltage assumes that IIC_EN=1, otherwise the internal circuit will have no effect on the IIC bus. SCLEN is often used to stretch the low voltage time to put the host into Wait state so that the slave has enough time to process the data.
1	SR	IIC slew rate control bit. 1: Slew rate control is disabled to adapt standard speed mode (100K). 0: Slew rate control is enabled for fast speed mode (400K). **Note:**SR=1 means slew rate control is off, the IO is adapted to 100Kbps communication.
0	IIC_EN	IIC operating enable bit 1: Normal operation of the IIC. 0: IIC does not work. **Note:**IICCON register is used for controlling the communication operation. **Note:** When it is necessary to pull down the clock line, i.e. WR_SCL_EN/RD_SCL_EN=1, before sending and receiving the last Byte data, the software should turn off the function of pulling down the clock line, i.e. WR_SCL_EN/RD_SCL_EN=0. After finishing sending and receiving the last Byte data, the software should turn on the function of writing to pull down the clock line. This operation can be adjusted according to whether the host is software or hardware, the processing time of interrupt handler is self-adjusted.

11.2.5 IIC Transmit/Receive Buffer Register (IICBUFFER)

Address: 0xFB

Bit No.	7	6	5	4	3	2	1	0
Symbol	IICBUFFER							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	IICBUFFER	IIC transmit/receive data cache register; in the case of RD_SCL_EN is 0, when the host reads out data, the data in IICBUFFER is sent to the slave transmit cache register in 2 clocks after the interrupt is generated as the data sent by the slave. Therefore, the IICBUFFER interrupt data is prepared before the interrupt is generated.

11.2.6 IIC IO Configuration Register (IIC_IO_SEL)

XRAM_SFR Address: 0x2049

Bit No.	7	6	5	4
Symbol	-	-	-	-
Read/Write	-	-	-	-
Po Initial Value	-	-	-	-
Bit No.	3	2	1	0
Symbol	IIC_AFIL_SEL	IIC_DFIL_SEL	IICSDA_IO_SEL	IICSCL_IO_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	1	0	0	0

Bit No.	Symbol	Description
3	IIC_AFIL_SEL	IIC IO analog filter selection enable. 1: Selects the analog filter function. 0: No analog filter function selected
2	IIC_DFIL_SEL	IIC IO digital filter selection enable 1: Selects the digital filter function. 0: No digital filter function selected
1	IICSDA_IO_SEL	IIC_SDA IO Selection. 0: PA1 1: PB6

0	IICSCL_IO_SEL	IIC_SCL IO Selection. 0: PA2 1: PB7

11.3 IIC Configuration Procedure

Figure 11.8 shows the IIC configuration flow chart.

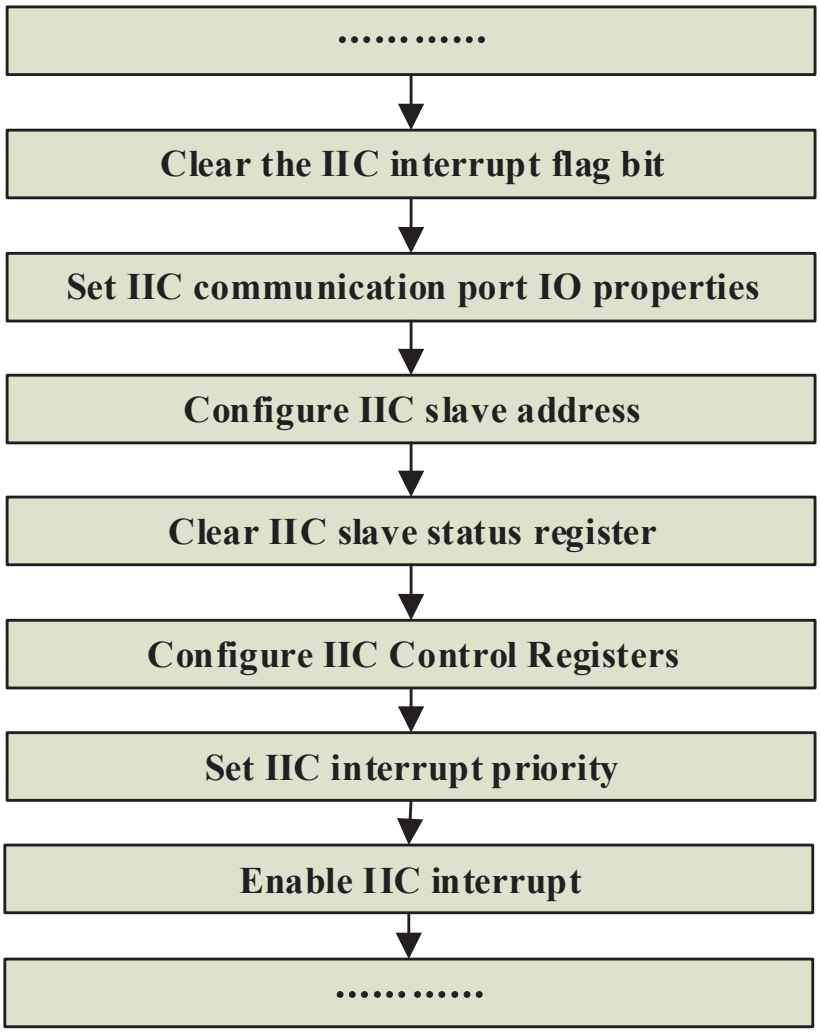


Table 0 .2 IIC Configuration

Note: IIC bus pull-up resistance is within 4.7k~10k, for ground filter capacitor it is recommended 10pF to 100pF near the pin.

12 SPI

SPI is a serial, synchronous, full-\half-duplex communication bus. The communication clock is a frequency division of the system clock. Under the 12M system clock frequency, the maximum frequency of the host supports 4M and the maximum frequency of the slave supports 2M. The communication mode has normal mode and high-speed mode. Clock idle level is available in four communication modes. The Duty Cycle of SPI clock is 50% (with 10% deviation allowed).

SPI normal mode: MCU writes into SPI transmit buffer SPID through interrupt (when SPI is enabled, it immediately generates a transmit empty interrupt) or polling, and the data is automatically loaded into the shift register and sent to SDO synchronously via SCLK; at the same time, the data is received from SDI and loaded into SPI receive buffer, and the receive data can be read from SPID when an accept full interrupt is generated.

SPI high-speed mode: The MCU writes transmit data (up to 2K) to SRAM in advance, and during communication, SPI reads out the data to be sent directly from SRAM without interrupt or polling; meanwhile, every time a byte of data (8Bits) is received, it is immediately written to the corresponding address in SRAM. When communication is completed, the SPI generates both the transmit empty flag and the receive full flag, and sends an interrupt.

The maximum operating frequency of the SPI host and slave modes is the bus clock divided by the baud rate of 2. The four modes are configured via SFR.

- CPOL: Select the clock idle state level. 0: clock idle state is low, 1: clock idle state is high.
- CPHA: Selects the data time for each cycle. 0: Data sampling at the first jump edge (rising or falling edge) of the clock. 1: Data sampling at the second jump edge (rising or falling edge) of the clock.

Mode 0 (CPOL=0, CPHA=0): clock idle level is low and the host and slave are sampling data on the rising edge.

Mode 1 (CPOL=0, CPHA=1): clock idle level is low, the host and slave are sampling data on the falling edge.

Mode 2 (CPOL=1, CPHA=0): clock idle level is high, the host and slave sample at data on falling edge.

Mode 3 (CPOL=1, CPHA=1): clock idle level is high, the host and slave are sampling data at rising edge.

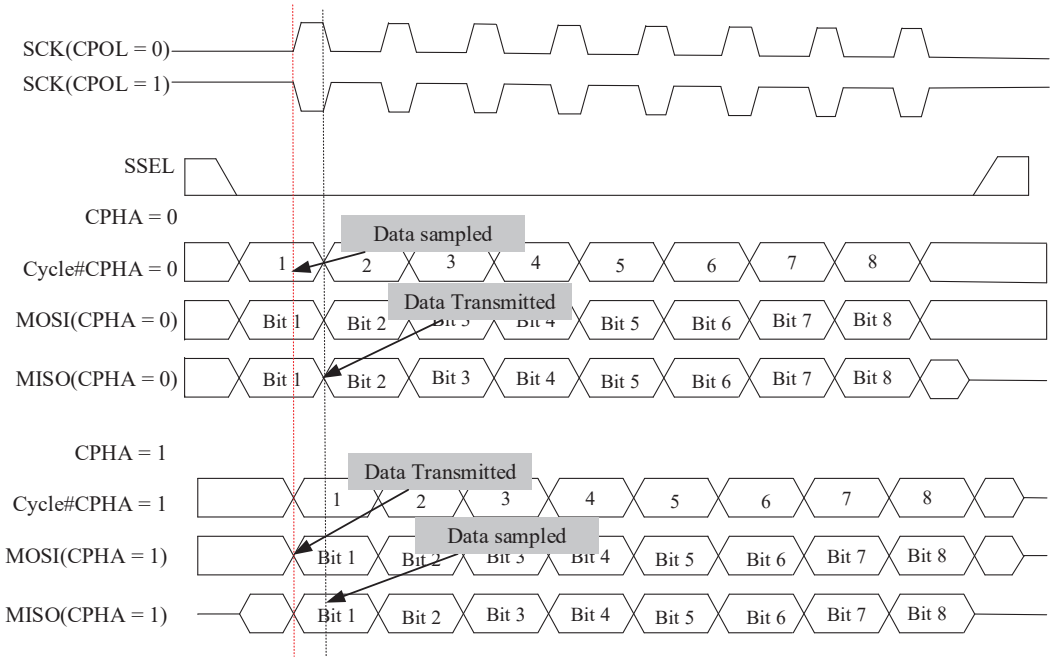


Table 12 .1 SPI operating mode sequence diagram

Descriptions:

SI: Slave sampling data; **SO**: Slave transmitting data;

MI: Host sampling data; **MO**:Host transmitting data.

The minimum time required for SPI_CS to become high is 1 spi_clk clock cycle.

12.1 Functions

1) **Host communication function:** configured as host via internal registers, select the corresponding off-chip slave for each 8bit data serial exchange, send interrupt to MCU when data transmission and receive is completed, MCU reads out data before the next byte of data transmission is completed, otherwise the current data overflows.
2) **Slave communication function:** configured as a slave via internal registers, waiting to be selected for 8bit (per time) data serial exchange with the host.
3) **Disable the system clock F_sys_clk, interrupt to wake up the CPU:** After turning off the core clock, the slave receives a full interrupt to wake up the core.

12.2 Communication Sequence

There are three flag bits, two interrupt mask bits, and an interrupt vector associated with the SPI system. The SPI interrupt enable bit (RX_IE) allows interrupts to occur from the SPI receiver full flag (SPRF). SPI transmit interrupt enable bit (TX_IE) allows interrupts to occur from the SPI transmit buffer empty flag (SPTEF). When a flag bit is set and the associated interrupt enable bit is set, a

hardware interrupt request is sent to the CPU. If the interrupt enable bit is cleared, software can poll the associated flag bit without an interrupt occurring. the SPI interrupt service routine (ISR) should check the flag bit to determine the event that caused the interrupt. The service routine should also clear the flag bits before returning from the ISR (usually near the start point of the ISR).
If there is new data in the Tx buffer, the host will send continuously, and ss_n_out cannot be pulled Up in the middle.

12.3 SPI Registers

12.3.1 SPI Configuration Register 1 (SPI_CFG1)

Address: 0xFC

Bit No.	7	6	5	4	3	2	1	0
Symbol	RX_IE	SPI_EN	TX_IE	MSTR	CPOL	CPHA	LSBFE	CS_N
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	1	0	1	0	1

Bit No.	Symbol	Description
7	RX_IE	Receive Enable - This is the SPI Receive Buffer Full (SPRF) interrupt enable 1: Interrupt enabled 0: Interrupt disabled (using polling)
6	SPI_EN	SPI Enable 1: Module enabled 0: Module disabled
5	TX_IE	Transmit Enable - This is the SPI Receive Buffer Full (SPTEF) interrupt enable 1: Interrupt enabled 0: Interrupt disabled (using polling)
4	MSTR	Host/Slave Mode Selection 1: Host mode 0: Slave mode
3	CPOL	SCLK active voltage selection 1: Active low 0: High level active
2	CPHA	SCLK phase selection. 1: transmit data on the first active clock edge

		0: samples data at the first active clock edge
1	LSBFE	LSB-first shifting (shifter direction) 1: SPI serial data transfer starts at the lowest bit 0: SPI serial data transfer starts at the highest bit
0	CS_N	The chip select signal control, CS is high by default. 0: Pull down CS 1: Pull up CS

Note 1:
In master mode, CPOL and CPHA must be configured before pulling down the chip select CS (the corresponding GPIO of CS must be configured before configuring the CS), otherwise SCLK_OUT has burrs.
Note 2:
In slave mode, after the chip select is pulled low, SPI_EN cannot be turned off, otherwise when SPI_EN is turned on again, there is a burr in the internally generated SCLK when the chip select becomes low again.

12.3.2 SPI Configuration Register 2 (SPI_CFG2)

Address: 0xFD

Bit No.	7	6	5	4
Symbol	-	FEEDBACK	HSPEED_START	HALF_DUPLEX
Read/Write	-	Read/Write	Read/Write	Read/Write
Po Initial Value	-	0	0	1
Bit No.	3	2	1	0
Symbol	BIDIR_SELECT	SPR		
Read/Write	Read/Write	Read/Write		
Po Initial Value	1	0		

Bit No.	Symbol	Description
6	FEEDBACK	Transmit the received data to the master/slave 1: Send the received data to the master\slave (the first byte of data sent is 0xff, the second is the first data received from the master, the later data sent are the previous data sent by the master. (Note: This function is only used in the mode of CPHA=0) 0:Transmit the data written by MCU to the master/slave
5	HSPEED_START	High-speed SPI communication mode enabled, work done, hardware automatically pulls down 1: High-speed SPI communication mode enabled 0: High-speed SPI communication mode disabled

		(In high-speed SPI mode, the chip select signal cannot be pulled up, whether in slave or master mode, which will result in loss of data sent by the SPI)
4	HALF_DUPLEX	Half-duplex mode selection 1: Select half-duplex mode 0: Select full-duplex mode
3	BIDIR_SELECT	Half-duplex mode, transmit and receive direction selection 1: Transmit 0: Receive
2~0	SPR	SPI baud rate factor: maximum communication frequency is 2M 0: F_sys_clk/2, 1: F_sys_clk/4, 2: F_sys_clk/6 3: f_sys_clk/8, 4: f_sys_clk/10, 5: f_sys_clk/12 6: F_sys_clk/14, 7: F_sys_clk/16

12.3.3 SPI Status Register (SPI_STATE)

Address: 0xFE

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	SPRF	OVERFLOW_RX	SPTEF
Read/Write	-	-	-	-	-	Read/Write	Read/Write	Read/Write
Po Initial Value	-	-	-	-	-	0	0	1

Bit No.	Symbol	Description
2	SPRF	Read buffer full flag, software write 0 to clear. 0: No data available in the receive data buffer 1: Data in the receive data buffer
1	OVERFLOW_RX	In the normal communication mode, when the read is not timely resulting in receive overflow, OVERFLOW_RX=1, this signal does not generate interrupt, only flag. Invalid in high-speed SPI communication mode (when the number of received data is equal to the configured {SPI_NUM_H,SPI_NUM_L}, it will end the operation and SPRF will be set, generating a full interrupt).
0	SPTEF	The transmit buffer empty flag, and hardware automatically clears by writing into the SPID. In SPI idle state, the first data written to the SPID will be stored directly into the shift register, and the second data written will be loaded into the transmit buffer, and SPTEF will be automatically pulled down.

12.3.4 SPI Data Register (SPI_SPID)

Address: 0xFF

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	--	Reading out the SPID register returns the data read from the receive data buffer rx_reg. When writing this register, the data will be written into the transmit data buffer tx_reg. Data should not be written to the transmit data buffer unless the SPI Transmit Buffer Empty Flag (SPTEF) is set, indicating that there is some space in the transmit buffer to queue new transmit bytes. Data can be read out from the SPID at any time after the SPRF is set and before another transmission is completed. Failure to read data from the receive data buffer before the end of a new transmission results in a receive overflow, the OVERFLOW_RX flag is set, and the newly transmitted data is lost.

When SPI_EN=0, this register cannot be written into.
Directly reading out the SPID register will return the data in current receive data buffer, when read the SPI_STATE status register first to determine SPRF=1, and then read the SPID, only then can clear the SPRF flag and correctly read out the data in receive data buffer.
The SPI_STATE status register is read out first to determine SPTEF=1 (transmit buffer empty flag) before the SPID register (transmit data buffer) can be written. When the SPI is configured as master, writing data to the transmit data buffer will initiate an SPI transfer.

12.3.5 SPI Transmit Buffer Start Address Register (SPI_TX_START_ADDR)

XRAM_SFR Address: 0x2057

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	--	In SPI high-speed mode, the first address of the transmit data cache SPI_TX_START_ADDR*8

12.3.6 SPI Receive Buffer Start Address Register (SPI_RX_START_ADDR)

XRAM_SFR Address: 0x2058

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	--	In SPI high-speed mode, the first address of the receive data cache SPI_RX_START_ADDR*8

12.3.7 SPI Data Buffer Address Number Low 8-bit Register (SPI_NUM_L)

XRAM_SFR Address: 0x2059

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	--	In SPI high-speed mode, the number of data cache addresses, low 8 bits

12.3.8 SPI Data Buffer Address Number High 3-bit Register (SPI_NUM_H)

XRAM_SFR Address: 0x205A

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	--		
Read/Write	-	-	-	-	-	Read/Write		

Po Initial Value	-	-	-	-	-	0

Bit No.	Symbol	Description
2~0	--	Number of data cache addresses in SPI high-speed mode, high 3 bits

12.3.9 SPI Host Mode Receiver Clock Selection Register (SPI_MCLK_MOD)

XRAM_SFR Address: 0x205B

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	-
Read/Write	-	-	-	-	-	-	-	Read/Write
Po Initial Value	-	-	-	-	-	-	-	0

Bit No.	Symbol	Description
0	-	SPI Master Mode Receiver Clock Select Register 1: Selects the master output as receiving clock 0: Selects PAD input for receiving clock

12.3.10 SPI IO High Speed Mode Enable Register (SPI_SPEED_EN)

XRAM_SFR Address: 0x205F

Bit No.	7	6	5	4
Symbol	-	-	HSPEED_EN_PC4	HSPEED_EN_PC3
Read/Write	-	-	Read/Write	Read/Write
Po Initial Value	-	-	0	0
Bit No.	3	2	1	0
Symbol	HSPEED_EN_PC2	HSPEED_EN_PA3	HSPEED_EN_PA2	HSPEED_EN_PA1
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
5	HSPEED_EN_PC4	SPI communication IO PC4 configured as high-speed mode enable. 1: High-speed mode. 0: Normal mode

4	HSPEED_EN_PC3	SPI communication IO PC3 configured as high-speed mode enable. 1: High-speed mode. 0: Normal mode
3	HSPEED_EN_PC2	SPI communication IO PC2 configured as high-speed enable. 1: High-speed mode. 0: Normal mode
2	HSPEED_EN_PA3	SPI communication IO PA3 configured as high-speed enable. 1: High-speed mode. 0: Normal mode
1	HSPEED_EN_PA2	SPI communication IO PA2 configured as high-speed enable. 1: High-speed mode. 0: Normal mode
0	HSPEED_EN_PA1	SPI communication IO PA1 configured as high-speed enable. 1: High-speed mode. 0: Normal mode

12.3.11 SPI IO Selection Register (SPI_IO_SEL)

XRAM_SFR Address: 0x2044

Bit No.	7	6	5	4
Symbol	-	-	-	-
Read/Write	-	-	-	-
Po Initial Value	-	-	-	-
Bit No.	3	2	1	0
Symbol	MISO_IO_SEL	MOSI_IO_SEL	SCK_IO_SEL	CS_IO_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
3	MISO_IO_SEL	MISO IO Selection. 0: PA2 1: PC3
2	MOSI_IO_SEL	MOSI IO Selection. 0: PA1 1: PC4
1	SCK_IO_SEL	SCK IO Selection. 0: PA3 1: PC2
0	CS_IO_SEL	CS IO Selection.

		0: PD0 1: PC5

Note:
By default, the chip select CS is high. In master mode, the corresponding GPIO must be configured before configuring the CS.

12.4 SPI Configuration Procedure

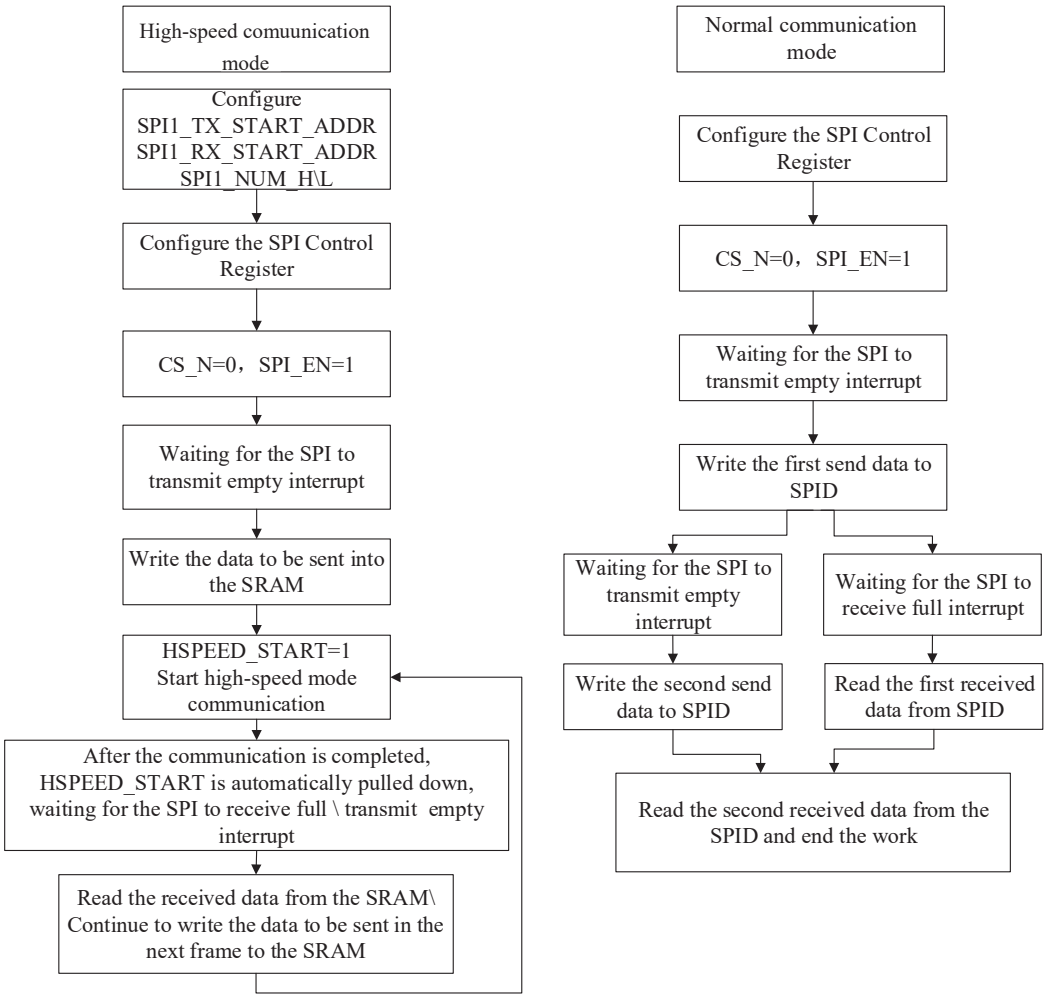


Figure 0.3 SPI Master/Slave Operation Flow Chart

Note:

1. Configure CPOL and CPHA when chip select CS is high, otherwise SCLK has glitches (Applied for both master and slave).

2. In slave mode, after the chip select is pulled down, SPI_EN cannot be turned off, otherwise there is a glitch in the internally generated SCLK when the chip select becomes low again when SPI_EN is turned on again. That is, SPI_EN cannot be turned off while SPI is selected.
3. In slave mode, if the chip select is always 0, if you want to switch CPOL/CPHA/LSBFE in the middle, the slave can switch only after the master pulls up the chip select.
4. In high speed mode, HSPEED_START will be automatically pulled down after work is done, at this time, the master cannot send SCLK again, otherwise it will generate an unsteady state.
5. In high-speed mode, if an odd number of data is sent in each frame, the chip select signal CS needs to be pulled high once between each frame.
6. The chip select signal CS must be pulled high once before switching from normal mode to high-speed mode.

13 UART

There is one UART module in the BS9000AMxx, supporting up to 5 IOs mapping, which can only correspond to one group of mappings at the same time. UART module IO features:

- Support full-duplex, half-duplex serial communication
- With independent double buffer receiver, single buffer transmitter
- Programmable baud rate (10-bit analog-to-digital frequency divider)
- Interrupt-driven or poll operation.
  - Transmit completed
  - Receive full
  - Receive overflow, parity error, frame error
- Support hardware parity generation and check
- Programmable 8-bit or 9-bit character length
- STOP bit 1 bit or 2 bits selectable
- Supports multi-processor mode
- Support TXD/RXD separate enable

13.1 Description

13.1.1 Baud Rate Generation

Baud rate generation modulo bandrate={UART_CON2.UART_BDH[1:0],UART_BDL}.
Baud rate calculation formula: when bandrate=0, no baud rate clock is generated, when bandrate=1~1023, SCI baud rate= BUSCLK/(16xbandrate). BUSCLK can select 24Mhz system clock or external oscillator.
Each time the baud rate register is configured the internal counter is cleared to regenerate the baud rate signal. Communication requires the transmitter and receiver to use the same baud rate. Allowable baud rate deviation range for communication: ±5%.

13.1.2 Functions of Emitter

Transmitting data process: write the UART_BUF to enable the transmitting, the transmit interrupt is set after the stop bit is sent, the software clears the interrupt flag and waits for the next writing.
The idle state of the transmitter output pin (TxD) defaults to the logic high state. The whole transmit process must be performed when the module is enabled (UART_CON1.UART_ENABLE=1) and the

transmitter is enabled (UART_CON1.TRANS_ENABLE=1).

By writing data into the data register (UART_BUF), it will store the data directly to the transmit data buffer and turn on the transmitting process. During the subsequent complete transmit process, writing to the data register UART_BUF and T8 is not allowed until the transmit completion stop bit is set and the transmit interrupt flag is set, then the UART_BUF can be written again to restart a new transmit.

The central component of a serial transmitter is a transmit shift register with a size of 10/11/12 bits (depending on the setting in the DATA_MODE control bit). Assuming DATA_MODE=0, the normal 8-bit data mode is selected. In 8-bit data mode, there are 1 start bit, 8 data bits and 1 to 2 stop bits in the shift register.

Transmit and receive are in little endian mode (LSB first).

When the transmitter is not enabled, the TXD IO is released and the corresponding PAD can be used for other functions.

13.1.3 Functions of Receiver

The receiver is enabled by setting the RECEIVE_ENABLE bit in the UART_CON1. Note that the whole receive process must be performed when the module is enabled, i.e. UART_CON1.UART_ENABLE=1.

When the receiver is not enabled, the RXD IO is released and the corresponding PAD can be used for other functions.

Receive data process: receive data at any time when receive enable is active, set the receive interrupt after receive stop bit, and software will clear the interrupt flag.

The current received data will have a detection mechanism, which can detect 3 kinds of errors: receive overflow, frame error, parity error, all of which need software to clear the flags. It is recommended to read out the status flag after detecting the receive interrupt, read out the data UART_BUF, and finally clear all the receive data status flags (UART_STATE[3:0]).

The data character consists of the start bit of logic 0, 8 (or 9) data bits (LSB-first), and the stop bit of logic 1 (1 bit). After receiving the stop bit into the receive shifter, if the receive data register is not yet full (RX_FULL_IF=0), the data character is transferred to the receive data register, setting the receive data register full (RX_FULL_IF=1) status flag. If RX_FULL_IF is already set at this time with the receive data register full, the overflow (RX_OVERFLOW_IF) status flag is set and new data is lost. Because the receiver is double buffered, the program has a full character time for reading after setting RX_FULL_IF and before reading the data from the receive data buffer to avoid receiver overflow.

When the program detects that the receive data register is full (RX_FULL_IF=1), it obtains the data from the receive data register by reading UART_BUF.

13.1.4 Receiver Sampling Method

The receiver uses a 16x baud rate clock for sampling. The receiver searches for falling edges on the RxD serial data input pins by extracting logic level samples at 16x baud rate. The falling edge is

defined as a logic 0 sample after 3 consecutive logic 1 samples. 16x baud rate clock is used to divide the bit times into 16 segments, labeled from RT1 to RT16.

Then the receiver samples each bit time of RT8, RT9 and RT10, including the start and stop bits, to determine the logic level of that bit. The logic level is the logic level of the majority of samples extracted during the bit time. When the falling edge is positioned, the logic level is 0 to ensure that this is the true start bit and not the noise. If at least two of these three samples are 0, the receiver assumes it is synchronized with the receiver character and begins shifting to receive the following data, and if the above conditions are not met then exits the state machine to return to wait for falling edge state.

The falling edge detection logic constantly looks for falling edges, and if an edge is detected, the sample clock resynchronizes the bit time. This improves receiver reliability when there is noise or a mismatch of baud rates.

13.1.5 Multi-processor Mode

Multiprocessor mode works in 9-bit mode only, the receive interrupt is set when the received UART_R8 bit = 1, otherwise it is not set. The effect of this mechanism is to eliminate the software overhead of processing unimportant message characters using hardware detection, allowing the receiver to ignore characters in messages used for different receivers.

In this application system, all receivers estimate the address character (bit 9 = 1) of each message, and once it is determined that the message is intended for a different receiver, subsequent data characters (bit 9 = 0) are not received.

Configuration process: configure receive enable, configure multiprocessor mode, receive address data (bit 9 = 1), receive and generate interrupt, used to confirm whether the address matches. If matching then configure to disable multiprocessor mode, all subsequent data (bit 9 = 0) can be received and generate an interrupt until the next received address data. If addresses do not match, then enable multiprocessor mode, all subsequent data are not be received until the next address data, and so on in a cyclic application.

Hardware response: transmit data, enabled by writing the UART_BUF value, set the transmit interrupt flag after the stop bit is sent, software clears the interrupt flag, and wait for the next write. Receive data is received at any time when receive enable is active, receive interrupt is set after receive stop bit, and software clears the interrupt flag. The currently received data will have a detection mechanism that detects three types of errors: receive overflow, frame error, and parity check error, all of which require software to clear the flags. It is recommended to read out the status flag after detecting the receive interrupt, and clear all the receive status flags UART_STATE[0:3].

Note: The mapping synchronization output function is not supported.

13.2 UART Register

13.2.1 UART Control Register 1 (UART_CON1)

Address: 0xBE

Bit No.	7	6	5	4
Symbol	UART_ENABLE	TRANS_ENABLE	RECEIVE_ENABLE	MULTI_MODE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	STOP_MODE	DATA_MODE	PARITY_EN	PARITY_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	UART_ENABLE	Module enable. 1: Enabled 0: Disabled
6	TRANS_ENABLE	Transmitter enable. 1: Enabled 0: Disabled
5	RECEIVE_ENABLE	Receiver enable. 1: Enabled 0: Disabled
4	MULTI_MODE	Multi-processor mode for multi-slave system where hardware automatically masks invalid information. 1: Enabled 0: Disabled
3	STOP_MODE	STOP bit size selection. 1: 2 bits 0: 1 bit
2	DATA_MODE	Data mode selection, unified control of transmitter and receiver 1: 9-bit mode 0: 8-bit mode
1	PARITY_EN	Parity check enable 1: Parity check enabled; 0: Parity check not enabled
0	PARITY_SEL	Parity check selection 1: Odd check; 0: Even check

13.2.2 UART Control Register 2 (UART_CON2)

Address: 0xBF

Bit No.	7	6	5	4
Symbol	-	-	-	
Read/Write	-	-	-	
Po Initial Value	-	-	-	
Bit No.	3	2	1	0
Symbol	TX_EMPTY_IE	RX_FULL_IE	UART_BDH	
Read/Write	Read/Write	Read/Write	Read/Write	
Po Initial Value	1	1	0	

Bit No.	Symbol	Description
3	TX_FINISH_IE	Transmit interrupt enable 1: Enabled 0: Disabled (for polling mode)
2	RX_FULL_IE	Receive interrupt enable. 1: Enabled 0: Disabled (for polling mode)
1~0	UART_BDH	Baud rate modulo divisor register high 2 bits

13.2.3 UART Status Flag Register (UART_STATE)

Address: 0xC0

Bit No.	7	6	5	4
Symbol	-	UART_R8	UART_T8	TX_FINISH_IF
Read/Write	-	Read	Read/Write	Read/Write
Po Initial Value	-	0	0	0
Bit No.	3	2	1	0
Symbol	RX_FULL_IF	RX_OVERFLOW_IF	FRAME_ERR_IF	PARITY_ERR_IF
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
6	UART_R8	The 9th data of the receiver, read-only

5	UART_T8	The 9th data of the transmitter, read-only when parity is enabled
4	TX_FINISH_IF	Transmit interrupt flag. 1: Transmit is finished. 0: Transmit is finished, software writes 0 to clear, while writing 1 is invalid
3	RX_FULL_IF	Receive interrupt flag. 1: Receiving cache is full 0: Receive cache is empty, software writes 0 to clear, while writing 1 is invalid
2	RX_OVERFLOW_IF	Receive overflow flag. 1: Receive overflow (new data lost) 0: No overflow, software writes 0 to clear, while writing 1 is invalid
1	FRAME_ERR_IF	Frame error flag. 1: Frame error detected 0: No frame error detected, software writes 0 to clear, while writing 1 is invalid
0	PARITY_ERR_IF	Parity error flag. 1: Receiver parity error 0: Parity check is correct, software writes 0 to clear, while writing 1 is invalid

Description:

8-bit mode: parity enable is invalid.

9-bit mode: When the parity bit is enabled, the 9th bit is the calculated parity bit; when it is disabled, the ninth bit is the T8 written into.

13.2.4 UART Data Register (UART_BUF)

Address: 0xC1

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	FF							

Bit No.	Symbol	Description
7~0	--	Data register. Read returns the contents of the read-only receive data buffer, write goes into the write-only transmit data buffer

13.2.5 UART Baud Rate Control Register (UART_BDL)

Address: 0xC2

Bit No.	7	6	5	4	3	2	1	0
Symbol	UART_BDL[7:0]							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	UART_BDL[7:0]	Baud rate control register baud rate modulo divisor register low 8 bits, bandrate={UART_BDH[1:0], UART_BDL}. When bandrate =0, no baud rate clock is generated, when bandrate =1~1023, baud rate = BUSCLK/(16x bandrate)

13.2.6 UART Clock Selection Register (UART_CLK_SEL)

XRAM_SFR Address: 0x205C

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	-	-
Read/Write	-	-	-	-	-	-	-	Read/Write
Po Initial Value								0

Bit No.	Symbol	Description
0	--	UART clock selection register 0: Selects 24Mhz clock 1: Selects clk_xtal

13.2.7 UART IO Selection Register (UART_IO_SEL)

XRAM_SFR Address: 0x2045

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	UART_RXD_SEL			UART_TXD_SEL		
Read/Write	-	-	Read/Write			Read/Write		
Po Initial Value	-	-	0			0		

Bit No.	Symbol	Description

5~3	UART_RXD_SEL	UART_RXD IO selection. 0: PA2; 1: PB1; 2: PB5; 3: PB7; 4: PC1
2~0	UART_TXD_SEL	UART_TXD IO selection 0: PA1; 1: PB2; 2: PB4; 3: PB6; 4: PC0

13.3 UART Configuration Procedure

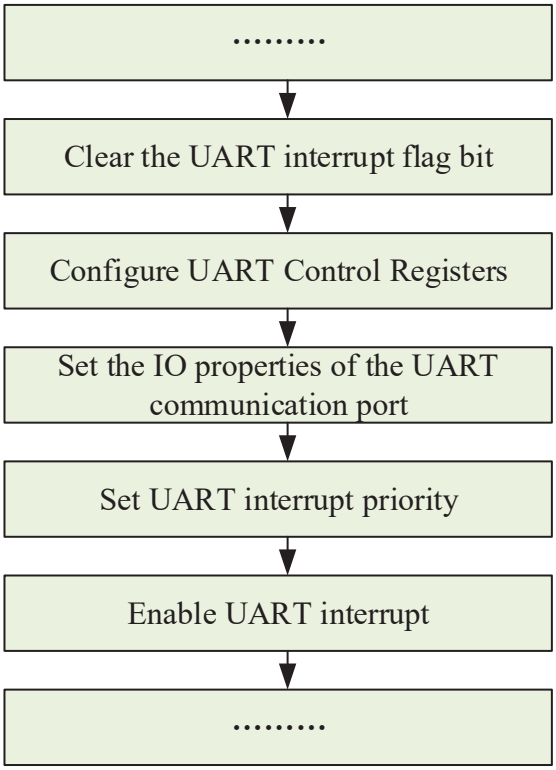


Table 13 .1 UART initialization configuration Flow Chart

Recommended application process:

1. Configure module enable, receive enable, mode selection: UART_CON1.
2. Configure baud rate and turn on interrupt enable: UART_BDL, UART_CON2.
3. Write into UART_BUF to start transmitting data and clearing the interrupt flag TX_FINISH_IF when a transmit interrupt is detected.
4. When the receive interrupt is detected, first read the receive status UART_STATE, then read UART_STATE.UART_R8 and UART_BUF, and finally clear the receive status flag (UART_STAT[3:0] = 0), when a receive process is finished, and wait for the next receive interrupt.
5. If configure interrupt is disabled, the program executes the UART, and also has to read the status flag first, then read out UART_STATE.UART_R8 and UART_BUF, and finally clear

the status flag.

6. Interrupt flag bit clear operation, in full-duplex operation, clear flag bit operation needs to write 0 for valid interrupt bits and 1 for other interrupt bits (write 1 is invalid operation), otherwise, it is easy to operate by mistake. For example, when the transmit interrupt is valid, you need to write UART_STATE = 0x0F; (i.e. configure UART_STATE[0:3] = 0x0F, UART_STATE.UART_R8 write invalid, UART_STATE.UART_T8 need to configure valid transmit data when it is in 9-bit mode and without parity check).
7. **8-bit mode**: parity enable is invalid.
8. **9-bit mode**: When the parity bit is enabled, the parity bit calculated by the ninth bit is not enabled, the ninth bit is written into UART_STATE.UART_T8. It only has transmit interrupt and receive interrupt, the error flag only marks the current data error detection, and only the corresponding bit is cleared by writing 0, no error interrupt, the transmit interrupt is set to 1 after the stop bit is sent and the software clears to 0. The receive interrupt is set to 1 after the stop bit is received, and the software clears to 0.

14 SCI

14.1 Introduction

The SCI module supports UART and LIN2.1 protocols.

14.1.1 UART Mode

- Support full-duplex/half-duplex serial communication
- Dual buffered transmitter and receiver with independent enable
- Programmable baud rate (13-bit analog-digital frequency divider)
- Interrupt-driven or polling operation.
  - Transmit data register empty, transmitting completed
  - Receive data register is full
  - Receive overflows, parity errors, frame errors and noise errors
  - Idle receiver detection
  - Active edge detection on the receive pin
- Support LIN synchronization interval segment detection
- Support hardware parity generation and checking
- Programmable 8-bit or 9-bit character length
- Programmable 1-bit or 2-bit STOP bit length
- Wake up receiver by idle line or address flag
- 13-bit abort character generation / 11-bit abort character detection optional
- Transmitter output polarity and receiver input polarity selectable
- Support baud rate adaptive function
- Support edge wake-up in wait mode and sleep mode

14.1.2 LIN Frame Mode

- Support Master and slave node transmission
- As master node, it supports automatic frame-header transmit, frame response transmit and receive
- As slave node, it supports automatic frame-header receive, frame response transmit and receive
- Support frame data storage, including PID, DATA, CHECKSUM
- Support classical checksum, enhanced checksum
- Support timeout detection
- Support Error detection
- Support baud rate detection

14.2 Baud Rate Generation

Baud rate generation modulo bandrate = {SCI_BDH[4:0], SCI_BDL}.
Baud rate calculation formula: when bandrate = 0, no baud rate clock is generated, when bandrate = 1 to 8191, SCI baud rate = BUSCLK/(16xbandrate). BUSCLK is SCI working clock, optional PLL_24MHZ clock or crystal clock XTAL.
Each time the baud rate register is configured, the internal counter is cleared to regenerate the baud rate signal.
Communication requires the transmitter and receiver to use the same baud rate.
Baud rate deviation range for communication can be 8/11*16=4.5%.
The system supports automatic baud rate matching, in LIN protocol, the synchronization segment character is 0x55. The baud rate detection is measured from the falling edge of the received START bit until the falling edge of the 8th data bit, 8 bits in total, and the bandrate will be updated automatically after the communication is finished, and can be read out through the register SCI_BDH/SCI_BDL. Note here that receiving the synchronization segment and automatically matching the baud rates is performed simultaneously with the receive function, and the receive full interrupt will be issued after the character is received. The maximum deviation before baud rate matching is not allowed to exceed 40%, otherwise the calibration fails.

14.3 Emitter Function

The idle state of the emitter output pin (TxD) defaults to the logic high state (SCI_C3.TXD_INV = 0 after reset). If SCI_C3.TXD_INV = 1, the emitter output is inverted.
The transmitter can send three types of characters: leading character, abort character, and data character. Writing 0 to SCI_C2.TRANS_ENABLE bit and then to 1 will make the leading character in the queue. Writing 1 to SCI_C2.BREAK_TRANS_START bit and then to 0 will make the abort character in the queue, write data register SCI_D will queue a data character.
The transmitter is enabled by setting the TRANS_ENABLE bit in SCI_C2. This queues the leading character, which is a complete character frame of the idle state, and controls the transmit of 12-bit or 11-bit or 10-bit idle characters (logic high) according to SCI_C1.DATA_MODE and SCI_C1.STOP_MODE. During normal applications where idle characters need to be sent, the program waits for the transmit null to be valid before setting it up to show that the last character of the message has moved to the transmit shifter and then writes 0 and 1 to the TRANS_ENABLE bit in that order. After that, as soon as the shifter is available, the operation queues the idle characters to be sent. Note that the SCI transmitter does not stop transmitting as long as the characters in the shifter (including the three characters) are not completed when TRANS_ENABLE=0.
By writing data to the SCI data register (SCI_D), the program stores data to the transmit data buffer that will queue a data character. The central component of the SCI transmitter is a transmit shift register of length 10 or 11 or 12 bits (depending on the setting in the SCI_C1. DATA_MODE and SCI_C1. STOP_MODE control bits). Assuming SCI_C1.DATA_MODE=0, the normal 8-bit data mode

is selected. In 8-bit data mode, there are 1 start bit, 8 data bits and 1 to 2 stop bits in the shift register. When the transmit shift register is available for a new SCI character, the value waiting in the transmit data register is transferred to the shift register while the transmit data register empty status flag (TX_EMPTY_IF) is set, indicating that another character can be written to the transmit data buffer of SCI_D.

An abort character is queued by writing a 1 followed by a 0 to the SCI_C2.BREAK_TRANS_START bit. The abort character is the full character time of logic 0 (10-bit time, including the start and stop bits). Longer abort characters of 13-bit time can be enabled by setting SCI_S2.BREAK_TRANS_SIZE=1. Similarly, both SCI_C1.DATA_MODE and SCI_C1.STOP_MODE can choose to increase one bit time. In general, the program waits for the transmit empty to be valid before setting it to show that the last character of the message has moved to the transmit shifter, and then writes 1 and 0 to the BREAK_TRANS_START bits in order. The operation then queues the abort characters to be sent as soon as the shifter is available. If SCI_C2.BREAK_TRANS_START is still 1 when an abort character already in the queue enters the shifter, additional abort characters will enter the queue.

If there are no new characters (including three types of characters) waiting in the transmit data buffer after stopping the shift out of the TxD pin, the transmitter sets the transmit finish flag and enters idle mode with TxD in high state waiting for more characters to be sent.

Note: The transmit data empty interrupt conditions include: configure the transmitter enable from 0 to 1 to turn on an empty interrupt. Also, enable an empty interrupt when transmitting the FIFO to the shift register. Turning off the transmitter enable during transmission will stop transmission after the current character is sent and will clear the previously queued characters.

The generation condition for a Transmit Finish Interrupt : enable a finish interrupt after all the queued characters are sent.

14.4 Receiver Function

The receiver is enabled by setting the RECEIVE_ENABLE bit in SCI_C2. The receiver input is inverted by setting SCI_C3.RXD_INV=1.

Receive characters include three types: data characters, abort characters, and idle characters.

The data character consists of the start bit of logic 0, 8 or 9 data bits (LSB first), and the stop bit of logic 1. After receiving the stop bit to the receive shifter, if the receive data register is not yet full (SCI_S1.RX_FULL_IF=0), the data character is transferred to the receive data register, setting the receive data register full (SCI_S1.RX_FULL_IF=1) status flag. If the SCI_S1.RX_FULL_IF at which the receive data register is full is already set at this time, the overflow (SCI_S1.RX_OVERFLOW_IF) status flag is set and the new data is lost. Because the SCI receiver is double buffered, the program has a full character time after setting SCI_S1.RX_FULL_IF and before reading out the data from the receive data buffer to avoid receiver overflow.

When the program detects that the receive data register is full (SCI_S1.RX_FULL_IF=1), it gets the data from the receive data register by reading SCI_D.

The abort character is counted from the 0 character of START until the STOP bit detects a 0 character,

and the SCI_S2.BREAK_CHECK_EN bit selects whether to enable 11-bit abort character detection, clearing the count once a rising edge is detected on the pin during the process. Enough 0 characters are detected (11 or 12 or 13 bits), set the abort character detection flag (SCI_S2.BREAK_CHECK_IF).

Idle character is selected according to SCI_C3.IDLE_SEL bit starting from the idle character bit count after the stop/start bit, and is detected only after the receiver has been active for a period of time (received full valid set once). Once a 0 character is detected the count is cleared and enough 1 characters (10/11/12 bits) are detected to set the idle character detection flag (SCI_S1.IDLE_IF).

Note: When abort character detection is enabled, only abort characters are detected and data receive is not considered, which is for LIN protocol flow control; when abort character detection is disabled, only data is received and abort character detection is ignored.

14.4.1 Receiver Sampling Method

The SCI receiver uses a 16x baud rate clock for sampling. The receiver searches for falling edges on the RxD serial data input pins by extracting logic level samples at 16x baud rate. The falling edge is defined as logic 0 sample after three consecutive logic 1 sampling. The 16x baud rate clock is used to divide the bit times into 16 segments, labeled from RT1 to RT16. When the falling edge is located, three more samples are extracted from RT3, RT5, and RT7 to ensure that this is the actual start bit and not just noise. If at least two of the three samples are 0, the receiver assumes it is synchronized with the receiver characters and starts shifting to receive the following data. If the above conditions are not met, then exit the state machine to return to wait for falling edge state.

The receiver then samples each bit time of RT8, RT9 and RT10, including the start and stop bits, to determine the logic level of the bit. The logic level is the logic level of the majority of the samples extracted during the bit time. Among the start bits, if at least 2 of the samples on RT3, RT5, and RT7 are 0, then that bit is assumed to be 0, even if one or all of the samples extracted on RT8, RT9, and RT10 are 1. If any sample in any bit time of the character frame (8 samples from RT3 to RT10 for the start bit and 3 samples RT8 to RT10 for the other bits) fails to match the the logical level of that bit, the noise error flag (SCI_S1.NOISE_ERR_IF) is set for all received characters when they are transmitted to the receive data buffer.

The falling edge detection logic constantly looks for falling edges, and if an edge is detected, the sample clock resynchronizes the bit time. This improves receiver reliability when there is noise or a mismatch of baud rates.

14.4.2 Receiver Wakeup from Sleep

Receiver wakeup from sleep is a hardware mechanism that uses hardware detection to eliminate the software overhead of handling unimportant message characters, allowing the SCI receiver to ignore characters in messages used for different SCI receivers.

In such applications, all receivers estimate the first character of each message, and as soon as they determine that the message is intended for a different receiver, they write logic 1 to the receiver wakeup (SCI_C2.RWU) control bit in SCI_C2. When the SCI_C2.RWU bit is set, the receiver-related status flags setting is not allowed (when the SCI_C3.RWU_IDLESEL bit is set, the idle bit IDLE is

set and an interrupt is generated).

In receiver sleep state (software sets the SCI_C2.RWU bit to enter sleep), the wake-up method can be selected via the SCI_C3.WAKE_SEL bit (i.e. hardware automatically clears the SCI_C2.RWU bit), including idle character wake-up and address flag wake-up.

Idle character detection is described above, and SCI_C2.RWU is automatically cleared once the receiver detects a full idle character. After wake-up, the receiver sets the corresponding status flag upon the next character received.

Address flag wakeup is when the receiver detects a logic 1 in the highest bits of the received characters (8th bit in SCI_C1.DATA_MODE=0 mode; 9th bit R8 in SCI_C2.DATA_MODE=1 mode), SCI_C2.RWU is automatically cleared.

After woken up, the receiver status flags and interrupts in the current character can be set.

14.5 Pin Connection Mode

When SCI_C1.CYCLE_MODE = 1, the SCI_C1.SINGLE_TXD bit selects either cyclic mode (SCI_C1.SINGLE_TXD= 0) or single line mode (SCI_C1.SINGLE_TXD= 1).

Cyclic mode

Cyclic mode is independent of external system connections and is sometimes used to check software to help isolate system problems. In this mode, the transmitter output is interconnected to the receiver input and the SCI does not use the RxD pin.

Single line mode

In single line mode, the SCI_C3.TXD_DIRECT bit controls the direction of the serial data on the TxD pin. When SCI_C3.TXD_DIRECT= 0, the TxD pin is an input to the SCI receiver, connecting to the receiver input, and when SCI_C3.TXD_DIRECT= 1, the TxD pin is an output driven by the transmitter.

14.6 Low power Wake-up

This module can be configured to enable the receive edge interrupt function to wake up the system in low power mode.

If the SCI module is configured to enable (SCI_C1.SCI_ENABLE=1), the receiver (SCI_C2.RECEIVE_ENABLE=1) and the RX edge interrupt (SCI_S2.RX_EDGE_IE=1) when the system enters sleep or wait mode, it will wake up the system when the receive IO is low .

14.7 Frame Mode

14.7.1 Frame Format

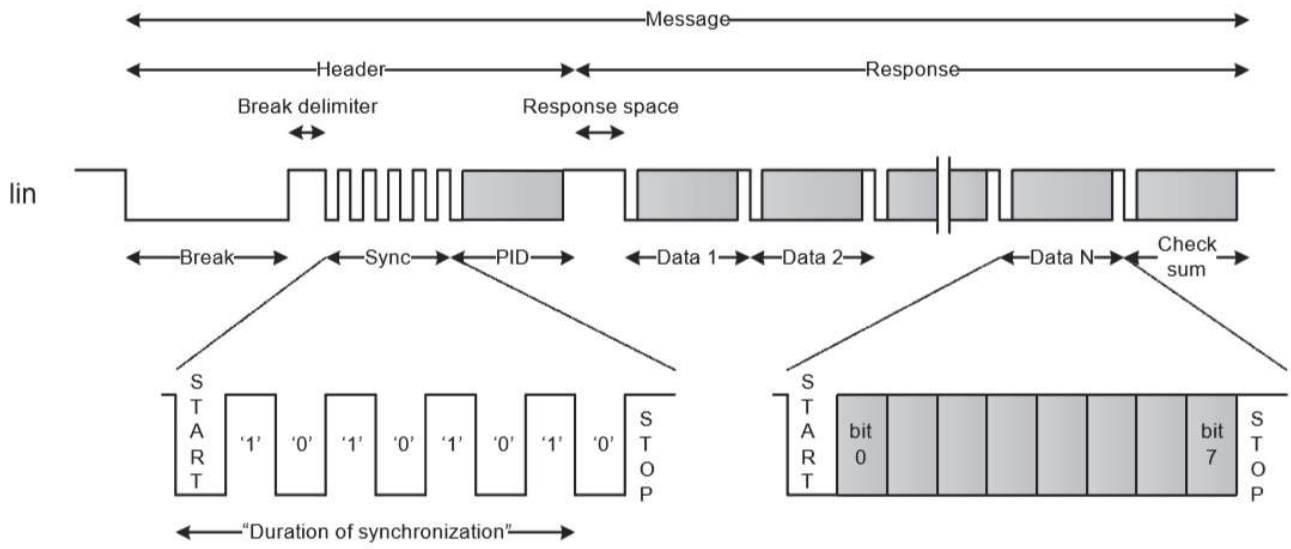


Table 14 .1 Frame Format

The master node sends the BREAK character according to the following table when the LIN_CMD.TX_HEADER command is valid (note that DATA_MODE here is fixed at 0 in frame mode). TX_HEADER_BREAK_ERROR is set when a bus "1" is detected at this stage.

When SCI_S2.BREAK_CHECK_EN is set, the slave node will decide to detect 11 or 12 bit "0" as BREAK character according to STOP_MODE, and set SCI_S2.BREAK_CHECK_IF at the same time.

BREAK_TRANS_SIZE	DATA_MODE	STOP_MODE	Length
0	0	0	10 bit times
0	0	1	11 bit times
0	1	0	11 bit times
0	1	1	12 bit times
1	0	0	13 bit times
1	0	1	14 bit times
1	1	0	14 bit times
1	1	1	15 bit times

Table 14 .2 Different Frame Lengths

14.7.1.1 BREAK Delimiter Segment

The master node determines the length of the BREAK interval character to be sent with 2 to 5 bytes selectable, according to LIN_CTL0.BREAK_DELIMITER_LEN. At this stage, if the bus is detected as “0”, then set the bit LIN_STATUS.TX_HEADER_BREAK_DELIMITER_ERROR.

14.7.1.2 SYNC Segment

The master node is fixed to transmit 0x55.
The slave node will be valid at LIN_CMD. RX_HEADER command, after detecting BREAK, fixed to receive 0x55, data is not stored, calculate the time from START falling edge to the falling edge of BIT7 (total 8 bits length), recorded in register LIN_SYNC_COUNTER, if there is no deviation, it should be 8*16=128. The deviation of 0.5% is allowed at the transmitter IO and 14% at the receiver IO as defined in the frame protocol, and the deviation range allowed for the synchronization segment is [106, 152] according to this calculation. If the deviation does not exceed this range, the slave node will automatically match the baud rates and continue the communication, and if the deviation exceeds this range, the interrupt flag LIN_INTR1.RX_HEADER_SYNC_ERROR will be set and the communication will be stopped.

14.7.1.3 PID Segment

The master node automatically computes the high 2 bits based on the configured 6-bit ID. LIN_INTR2.TX_HEADER_DONE is set after the frame header is sent. If there is no LIN_CMD.TX_RESPONSE or LIN_CMD.RX_RESPONSE command at this time, LIN_CMD.TX_HEADER will be cleared automatically.
bit6:P0 = ID0⊕ ID1⊕ ID2⊕ ID4
bit7:P1 = ~ (ID1⊕ID3⊕ID4⊕ID5)
The slave node will check automatically after the receive is finished and set LIN_INTR1.RX_HEADER_PARITY_ERROR for errors and LIN_INTR2.RX_HEADER_DONE is set when the receive is finished.

14.7.1.4 DATA Segment

This stage does not distinguish between master and slave nodes, instead it is distinguished by instruction. From the LIN_DATA_IDX register, it reads out how many bytes have been transferred.
LIN_CMD. TX_RESPONSE is the transmit response, the data sent (TX_DATA/TX_CHECKSUM) and the number of data (LIN_CTL0.DATA_NR) are configured before the command is enabled.
LIN_CMD.RX_RESPONSE is the receive response. The number of data received (LIN_CTL0.DATA_NR) and the checksum type (LIN_CTL0.CHECKSUM_ENHANCED) have to be configured before the STOP bit of the first data is issued, otherwise there may be a conflict.
In general, for the master node, the number of data and the checksum type of the response are configured before the frame transmission starts. Slave nodes need to decide these based on the received PIDs.

14.7.1.5 CHECKSUM Segment

Transmit Response Node will transmit checksum bytes at the same time, and the data is stored in

the transmit cache, which is obtained by the software based on the calculation.

The receive response node receives both checksum bytes and automatically checks the data. The checksum can be divided into an enhanced checksum with PID and a classical checksum without PID. If the checksum fails, the error flag LIN_INTR2.RX_RESPONSE_CHECKSUM_ERROR is set.

The checksum method is to add each byte of the checksum object in rounded binary (subtract 255 whenever the result is greater than or equal to 256), and invert the final sum bit by bit, and use the result as the checksum to be sent. The receiver conducts the same rounded binary addition to the received data according to the checksum type, the final sum is not inverted, and the sum is added to the received checksum. If the result is 0xFF, the checksum is correct, which ensures the correctness of data transmission to some extent.

14.7.1.6 TIMEOUT Handling

Three modes of timeout detection.

1) Full Frame Mode
2) Frame header mode
3) Frame Response Mode

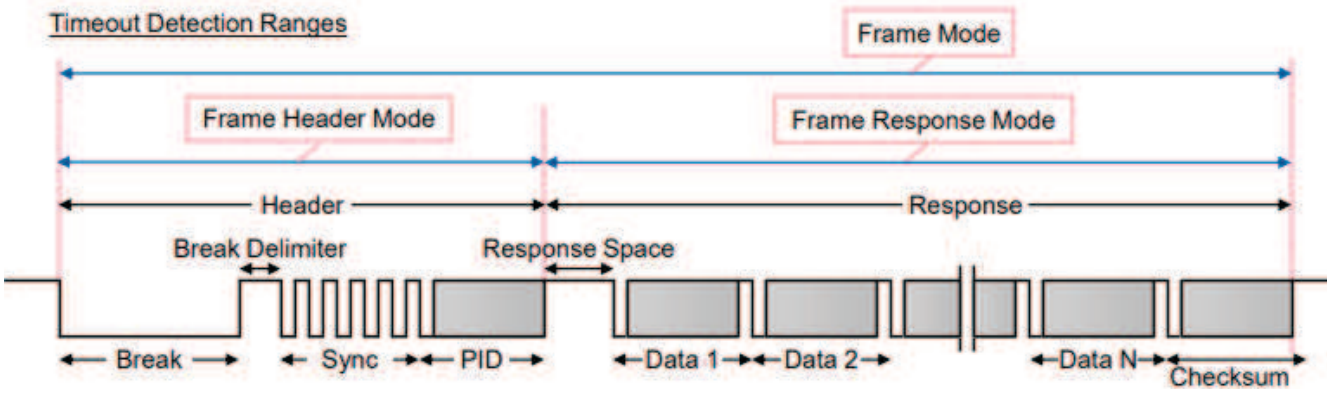


Table 14 .3 Frame timeout mode

According to register LIN_CTL0.FRAME_TIMEOUT_SEL, three timing methods can be selected. When the frame communication time exceeds the value configured in register LIN_FRAME_TIMEOUT, the error flag LIN_INTR1.TIMEOUT is set.

FRAME_TIMEOUT_SEL	Timeout selection	Timer start	Timer STOP
0	Timerout disabled	None	None
1	Frame mode	Falling edge of START bit in BREAK field	Checksum field STOP bit OR timeout

2	Frame header mode	Falling edge of START bit in BREAK field	PID field STOP bit OR timeout
3	Frame response mode	End of STOP bit	Checksum field STOP bit OR timeout

Table 14 .4 Frame Timeout Timing Methods

14.7.2 LIN Communication

The LIN protocol supports three types of information transfer.

1) Master response: the master node sends the frame header and sends a response. This type is used to observe and control slave nodes.
2) Slave response: the master node sends the frame header, the slave node sends a response, and the master node receives the response. This type is used to observe the state of the slave node.
3) Slave to slave: the master node sends the frame header, one slave node sends a response, and the other slave node receives the response.

To support these different message types, the manipulation of LIN master or slave operation modes is done indirectly by command sequences.

LIN_CMD.TX_HEADER: The master node sends a full frame header with this command, including the interval segment, the synchronization segment and the PID field.
LIN_CMD.RX_HEADER: The slave node receives the frame header with this command. After the slave node receives the frame header, the finish flag (LIN_INTR2.RX_HEADER_DONE) is set and then the slave node application can decide the following actions based on the received PID field.

- Receive a response (LIN_CMD.RX_RESPONSE).
- Transmit a response (LIN_CMD. TX_RESPONSE).
- Ignore a response, close the channel and re-enable the next frame

LIN_CMD.TX_RESPONSE: The master or slave node uses this command to transmit a response, the hardware sends the data field and the auto-generated checksum.
LIN_CMD.RX_RESPONSE: The master or slave node uses this command to receive a response, and the hardware receive data field stored in the cache to verify the checksum.

LIN Master Node Command Sequence

Massage types	Command Sequence in register LIN_CMD			
	TX_HEADER	RX_HEADER	TX_RESPONSE	RX_RESPONSE
Master Response	1	0	1	0
Slave Response	1	0	0	1
Slave-to-Slave	1	0	0	0

Response(transmitting node)				

Table 14 .5 LIN master node command sequence

For the master node, the order of these commands is set at the beginning of the frame.
To enter the relevant interrupt handling function only once, the Master node needs to enable an interrupt source LIN_INTR2.(TX_HEADER_DONE,TX_RESPONSE_DONE).

LIN Slave Node Command Sequence

Massage types	Command Sequence in register LIN_CMD			
	TX_HEADER	RX_HEADER	TX_RESPONSE	RX_RESPONSE
Master Response	0	1	0	1
Slave Response	0	1	1	1
Slave-to-Slave Response(transmitting node)	0	1	1	1
Slave-to-Slave Response(receiving node)	0	1	0	1
Ignore Response	0	1	0	0

Table 14 .6 LIN slave node command sequence

For slave nodes, RX_HEADER and RX_RESPONSE are to be set before detecting the interval segment, which can avoid losing the interval segment and the data field (because it is not sure what state the bus is currently in.) When both TX_RESPONSE and RX_RESPONSE are set, priority is given to TX_RESPONSE.
The slave node always sets two commands LIN_CMD. RX_HEADER and LIN_CMD. RX_RESPONSE, and the received frame header PID field will specify the slave node:

- Receive a response
- Transmit a response
- Abort transmission and ignore a response

By setting both LIN_CMD. RX_HEADER and LIN_CMD. RX_RESPONSE, the slave node responds first to the receive in order to avoid the loss of data fields in the response.
When a message is successfully transmitted, the command is cleared and re-enabled for the next transmission. On detection of an error, the transmit command is cleared, but the receive command is not cleared. This behavior is important for slave nodes in practical applications where interval segments occur during the receiving.
LIN_CMD.TX_RESPONSE and LIN_CMD.RX_RESPONSE can be enabled at the same time and the command order is handled according to the following priority.

- Highest priority: LIN_CMD.TX_RESPONSE command
- Second priority: LIN_CMD. RX_RESPONSE command

➢ Lowest priority: no response without these two commands

14.7.3 LIN Software Flow Chart

Figure 14.7 shows the master node communication software flow chart.

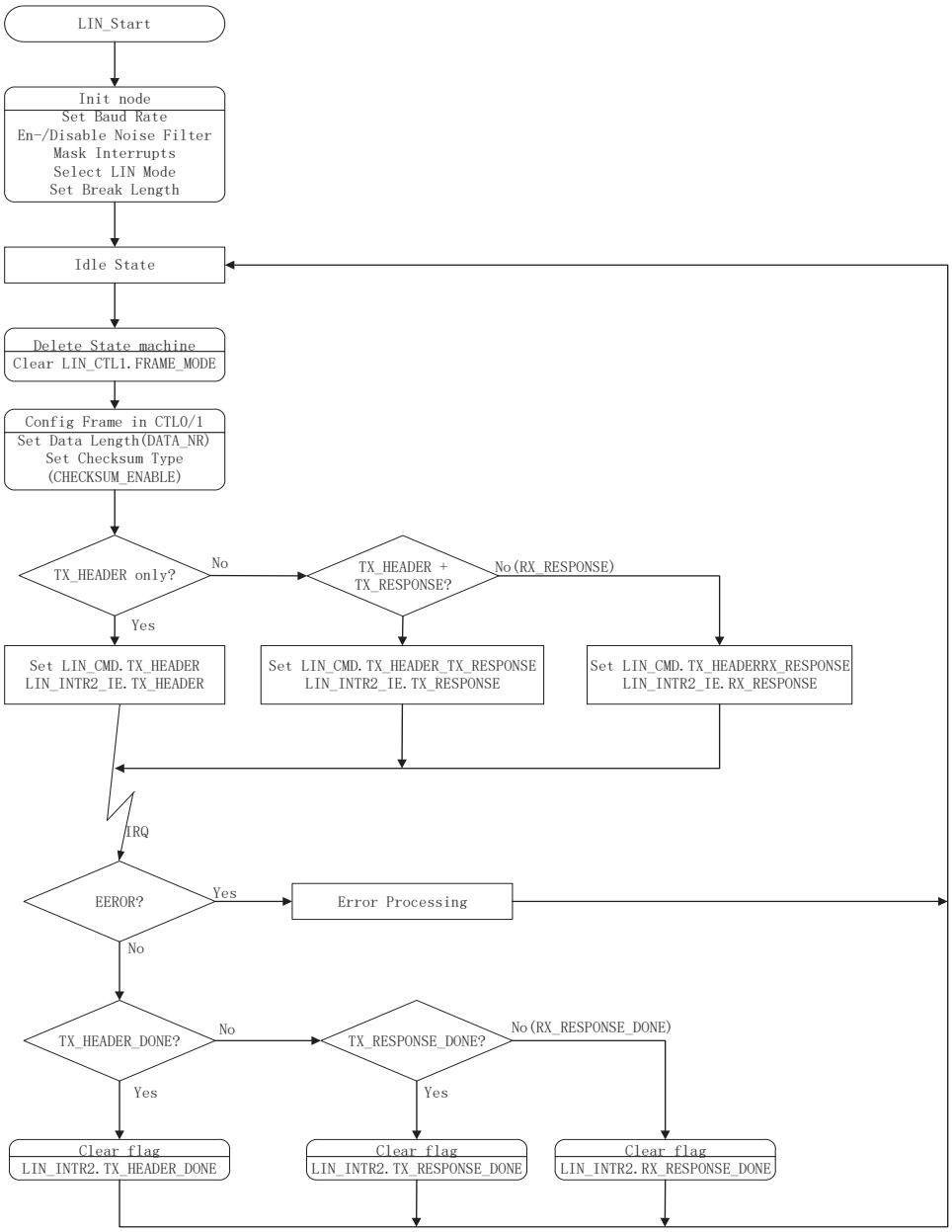


Table 14 .7 Master node communication software Flow Chart

Figure 14.8 shows the flow chart of the slave node communication software.

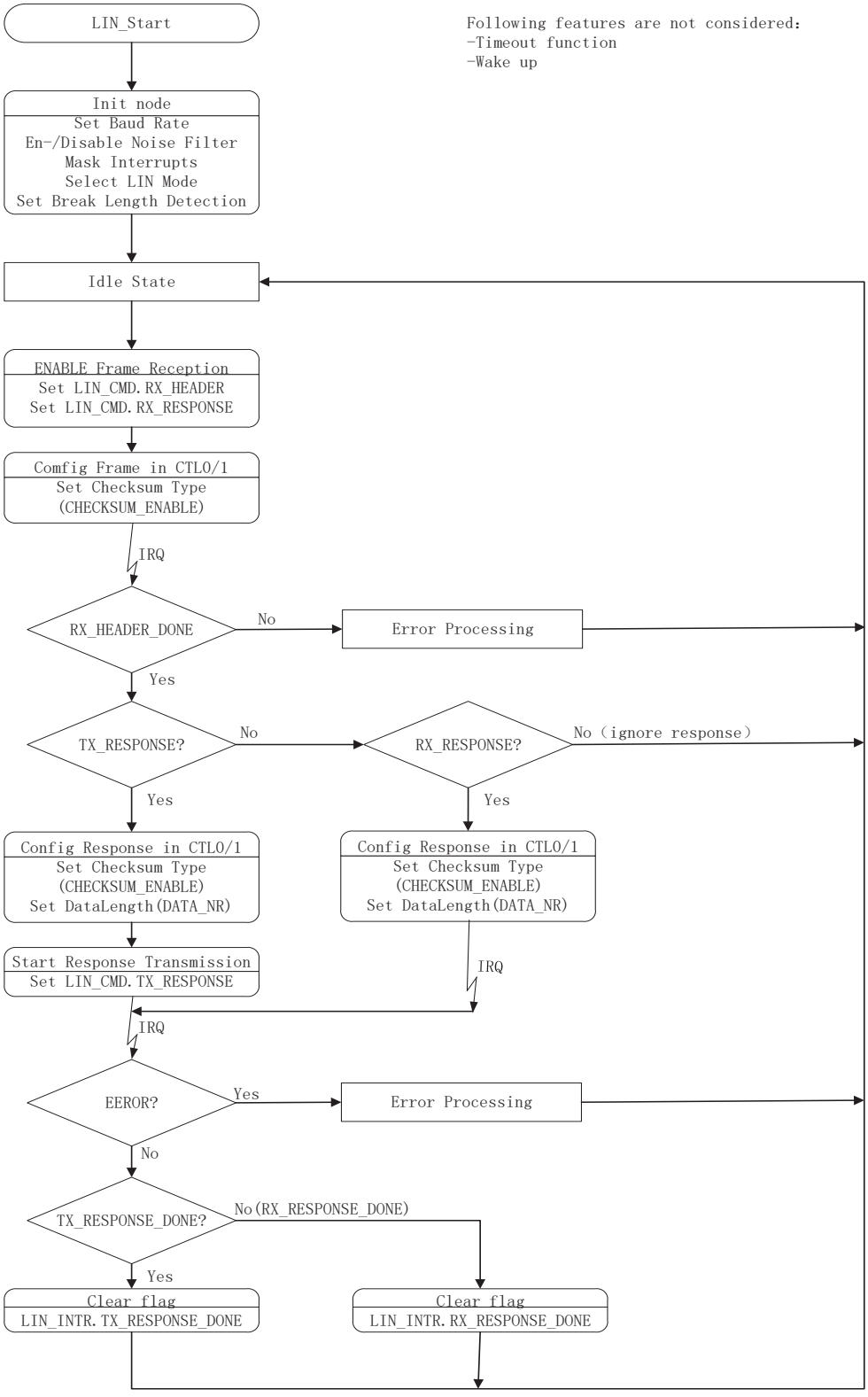


Table 14 .8 Slave Node Communication Software Flow Chart

14.8 Frame Mode State Machine

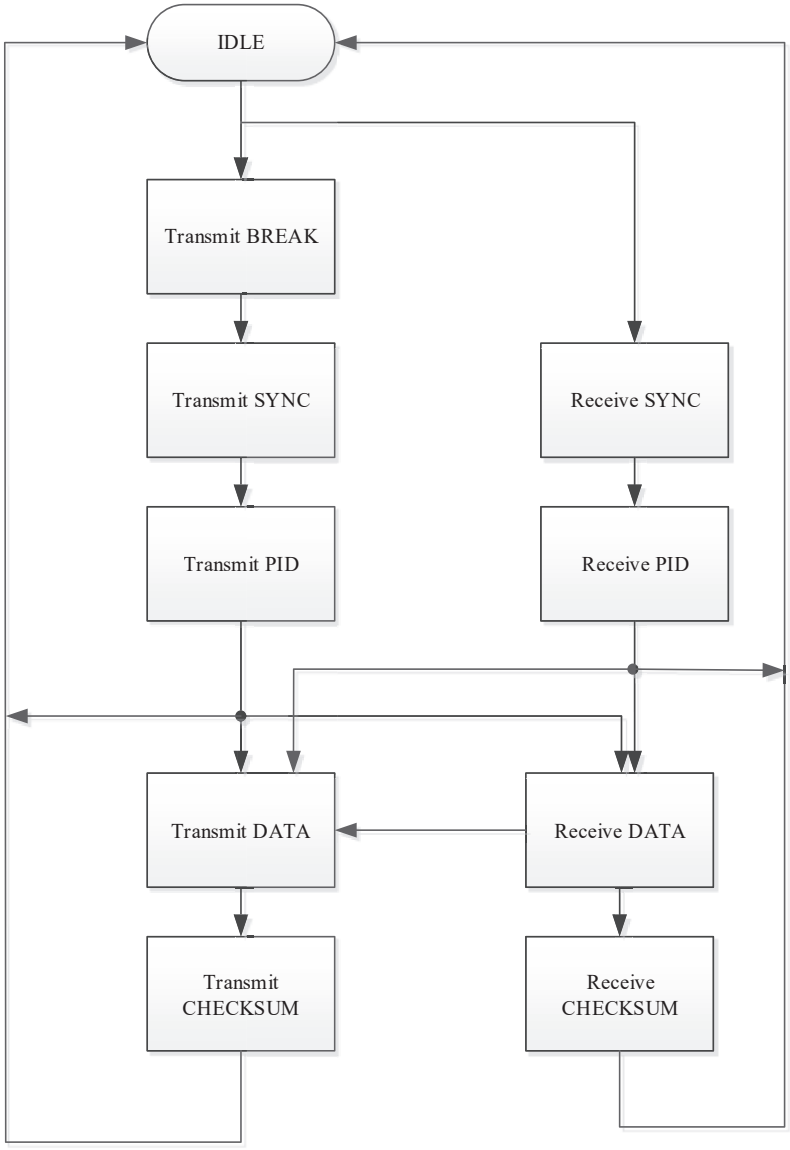


Table 14 .9 Frame mode operating state machine

Descriptions of each state machine.

- IDLE: Wait state.
- Transmit BREAK: Transmits the BREAK character (13 bits), including the BREAK interval (configurable).
- Transmit SYNC: Transmits synchronization character, 0x55.
- Transmit PID: Transmits PID character, 1 byte (two parity bits are calculated automatically).
- Transmit DATA: Transmits data character, 1~8 bytes (configurable).
- Transmit CHECKSUM: Transmits checksum character, 1 byte.

- Receive SYNC: Receives synchronous segment, baud rate auto-matching, data is not stored.
- Receive PID: Receives PID character, 1 byte.
- Receive DATA: Receives data characters, 1~8 bytes (configurable).
- Receive CHECKSUM: Receives checksum character, 1 byte.

14.9 Instructions for interrupt handling

There are two interrupt groups based on the working mode:

- UART mode: There are send/receive interrupts.
  - Send interrupts include TX_EMPTY, TX_FINISH.
  - Receive interrupts include RX_FULL, IDLE, BREAK_CHEK, and RX_EDGE.
  - Error flags RX_OVERFLOW, NOISE_ERR, FRAME_ERR, PARITY_ERR do not trigger the interrupt.
- LIN mode: There are BREAK_CHEK, RX_EDGE and LIN communication interrupt.LIN communication interrupts include normal interrupts and error interrupts.
  - Normal interrupt: TX_HEADER_DONE，TX_RESPONSE_DONE，RX_HEADER_DONE，RX_RESPONSE_DONE，RX_HEADER_SYNC_DONE.
  - Error interrupt: TIMEOUT, TX_HEADER_BIT_ERROR, TX_RESPONSE_BIT_ERROR, RX_HEADER_FRAME_ERROR, RX_HEADER_SYNC_ERROR, RX_HEADER_PARITY_ERROR, RX_RESPONSE_FRAME_ERROR, RX_RESPONSE_CHECKSUM_ERROR.

Note:

Note 1: In the same set of interrupts, enabled interrupts are processed by OR operation, so when multiple interrupts are enabled at the same time, once one interrupt is set, the interrupt handler function will not be entered again without being cleared.

Note 2: some of LIN error interrupt will abort the current frame transmission immediately , after the error normal send/receive interrupt will not occcur after the error occurred, including: TX_HEADER_BIT_ERROR (configurable), TX_RESPONSE_BIT_ERROR (configurable), RX_HEADER_FRAME_ERROR, RX_HEADER_SYNC_ERROR, RX_RESPONSE_FRAME_ERROR; Some errors occur at points that lag behind normal receive interrupts, including: RX_HEADER_PARITY_ERROR (receiving the frame header is complete and will abort the subsequent response transmission of the current frame), RX_RESPONSE_CHEGKSUM_ERROR (receiving the frame response is complete), it is recommended to handle this error in receive interrupts.

Note 3: TIMEOUT interrupt will not stop frame transmission. The interrupt handling should restart SCI, and the steps are as follows:

Clear the error flag bit (LIN_INTR1.TIMEOUT) after entering the TIMEOUT interrupt, clear the frame command (Clear LIN_CMD to 0), and disable the enablement module (Clear SCI_C1.SCI_ENABLE 0); To transfer the next frame, enable the module (Clear SCI_C1.SCI_ENABLE

0), reconfigure all frame related configuration registers (LIN_FRAME_TIMEOUT, LIN_CTL1, LIN_CTL0, LIN_INTR1_IE, LIN_INTR2_IE), Finally, configure LIN_CMD to enable communication.

14.10 SCI Registers

LIN_FRAME_TIMEOUT, LIN_CTL1, LIN_CTL0, LIN_INTR1_IE and LIN_INTR2_IE registers can be modified the configuration after enabling, but they cannot be configured continuously. If continuous configuration is required, it is recommended that the interval be more than 10 system clocks.

14.10.1 SCI Clock Selection Register (SCI_CLK_SEL)

XRAM_SFR Address: 0x205D

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							-
Read/Write	-							Read/Write
Po Initial Value	-							0

Bit No.	Symbol	Description
0	--	SCI Clock Select Register 0: Selects PLL_24Mhz clock 1: Selects clk_xtal

This register is used to select the module clock source, on which the baud rate generator is also based.
This register is also valid in LIN frame mode.

14.10.2 Baud Rate Control High 5-bit Register (SCI_BDH)

XRAM_SFR Address: 0x2070

Bit No.	7	6	5	4	3	2	1	0
Symbol	-			-				
Read/Write	-			Read/Write				
Po Initial Value	-			0				

Bit No.	Symbol	Description
4~0	--	Baud rate control register. bit[4:0]:Baud rate modulo divisor register high 5 bits

14.10.3 Baud Rate Control Low 8-bit Register(SCI_BDL)

XRAM_SFR Address: 0x2071

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	--	Baud rate control register. Baud rate modulus divisor register low 8 bits.

bandrate={SCI_BDH[4:0],SCI_BDL}, no baud rate clock is generated when bandrate=0. When bandrate=1~8191, SCI baud rate= BUSCLK/(16xbandrate).
This register is also valid in LIN frame mode.

14.10.4 Control Register 1 (SCI_C1)

Address: 0xB5

Bit No.	7	6	5	4
Symbol	CYCLE_MODE	STOP_MODE	SINGLE_TXD	DATA_MODE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	PARITY_EN	PARITY_SEL	RATE_MATCH_EN	SCI_ENABLE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	CYCLE_MODE	Cycle mode enable. 1: Cyclic mode or single line mode, txd connected to rxd 0: Normal two-line mode
6	STOP_MODE	Stop digit selection. 1: 2bits. 0: 1bit.
5	SINGLE_TXD	Single line mode enable. 1: Single line mode selected at CYCLE_MODE =1, txd pin valid

		0: Internal cycle mode, txd pin invalid
4	DATA_MODE	Transfer data mode selection. 1: 9-bit mode (the 9th bit is the parity bit) 0: 8-bit mode
3	PARITY_EN	Parity check enable. 1: Parity check enabled. 0: Parity disabled
2	PARITY_SEL	Parity check selection. 1: Odd check. 0: Even check
1	RATE_MATCH_EN	Synchronization segment (0x55) baud rate auto-match enable. 1: Adaptive baud rate update. 0: Fixed configuration baud rate
0	SCI_ENABLE	Module operating clock gate enable. 1; enable is valid, turn on the module working clock 0: Turn off the module working clock and reset the function module

The SINGLE_TXD bit is meaningful when CYCLE_MODE = 1 and is used to select whether the txd pin is active or not.

STOP_MODE: STOP bit selection, it is valid for both transmitter and receiver.

Symbol	Description
DATA_MODE	Transmission data mode selection, it is valid for both transmitter and receiver. For 1:9-bit mode, start + 8 data bits (LSB first sent) + 9th data bit + stop, the 9th bit is normally used as a parity bit. For 0:8-bit mode, start + 8 data bits (LSB first sent) + stop.
PARITY_EN	Parity check enable bit, which is used for receiver parity check enable and transmitter parity check generation enable, applicable in 8-bit/9-bit mode. In 8-bit mode transmitter sends the 8th bit for the first 7 bits of parity check generation, the receiver checks the first 7 bits to get the value compared with the 8th bit; in 9-bit mode transmitter sends the T8 for the 8-bit data calculation to generate the bit, the receiver checks the first 8 bits to get the value compared with the 8th bit. The receiver checks the first 8 bits of data to get the value compared with R8.

PARITY_SEL	Parity selection. Parity indicates that the total number of 1's in the data character (including parity bits) is odd. Even indicates that the total number of 1's in the data character (including parity bits) is even.
RATE_MATCH_EN:	Synchronization segment baud rate auto-match only applies to 8-bit data 0x55, that is, the next communication must be 0x55, it will automatically update the bandrate after the communication is finished, and can be read out through register SCI_BDH/SCI_BDL, can only synchronize once upon enabled; if need to synchronize again, need to pull the enable bit down once and pull it up again.
SCI_ENABLE	To use all the functions of SCI, first configure to enable the module with SCI_ENABLE=1 to turn on the operating clock. In the case of turning off the module enable, the module can only perform write/read registers.
CYCLE_MODE/ STOP_MODE /SINGLE_TXD/SCI_ENABLE	CYCLE_MODE/STOP_MODE SINGLE_TXD/SCI_ENABLE registers are also valid in LIN frame mode.

14.10.5 Control Register 2 (SCI_C2)

Address: 0xB6

Bit No.	7	6	5	4
Symbol	TX_EMPTY_IE	TX_FINISH_IE	RX_FULL_IE	IDLE_IE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	TRANS_ENABLE	RECEIVE_ENABLE	RWU	BREAK_TRANS_START
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	TX_EMPTY_IE	Transmit buffer empty interrupt enable.

		1: Interrupt enabled. 0: Interrupt disabled
6	TX_FINISH_IE	Transmit finish interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled
5	RX_FULL_IE	Receive full interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled
4	IDLE_IE	Idle line interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled
3	TRANS_ENABLE	Transmitter enable bit. 1: The transmitter is turned on. 0: Transmitter is off
2	RECEIVE_ENABLE	Receiver enable. 1: The receiver is turned on. 0: Receiver is off
1	RWU	Receiver wake-up control. 1: The receiver is in standby mode, waiting for the wake-up condition. 0: Receiver operates normally.
0	BREAK_TRANS_START	Transmit interval segment, write 1 and 0 to this bit successively, i.e. a interval segment is scheduled in the transmit data stream.

The interrupt enable register, which is set 1 to enable interrupt generation and set to 0 to disable interrupt generation, but does not affect the interrupt flag set in the following register, used for polling mode.

Symbol	Description
TRANS_ENABLE	Transmitter enable. This bit must be set to 1 in order to use the SCI transmitter. An idle character can be queued by clearing 0 to this bit and then setting 1.
RECEIVE_ENABLE	Receiver enable. To use SCI receiver, this bit must be set to 1.
RWU	RWU receiver wake-up control. This bit writes a 1 to put the SCI receiver to sleep state and wait the selected wake-up condition to be detected by hardware automatically. The wake-up condition can be either an idle line between messages or a logic 1 in the highest data bit in a character. Application

	software sets RWU or software clears it. General application hardware wake-up automatically clears RWU.
BREAK_TRANS_START	Transmit interval segment control, write 1 and 0 to this bit successively, i.e., a interval is scheduled in the transmit data stream, and a set of abort characters is sent once the shifter is available.
TRANS_ENABLE/RECEIVE_ENABLE/RWU	TRANS_ENABLE/RECEIVE_ENABLE/RWU registers are also valid in LIN frame mode.

14.10.6 Control Register 3 (SCI_C3)

Address: 0xB9

Bit No.	7	6	5	4
Symbol	R8	T8	TXD_DIRECT	TXD_INV
Read/Write	Read	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	RXD_INV	RWU_IDLESEL	IDLE_SEL	WAKE_SEL
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	R8	The 9th data of the receiver. Read only.
6	T8	The 9th data of the transmitter
5	TXD_DIRECT	txd pin direction selection in Single line mode. 1: TxD pin is output in single line mode 0: TxD pin is input in single line mode
4	TXD_INV	Invert the data at the txd. 1: The transmit data is reversed. 0: the transmit data is not reversed
3	RXD_INV	Invert the data at the rxd. 1: The received data is reversed. 0: Received data not reversed
2	RWU_IDLESEL	Receive wake-up from Idle detection. 1: Set the IDLE bit when an idle character is detected during the receive standby state (RWU=1).

		0: No IDLE bit is set when an idle character is detected during the receive standby state (RWU=1)
1	IDLE_SEL	Idle line type selection. 1: Idle character bit count starts after the stop bit 0: Idle character bit count starts after the start bit, counting 10-bit times (if DATA_MODE = 1 or STOP_MODE = 1, add 1 bit time, respectively)
0	WAKE_SEL	Receiver wake-up method selection. 1: Address flag wake-up. 0: Idle line wake-up

Symbol	Description
R8	The 9th data bit of the receiver, when configured for 9-bit data, R8 can be regarded as the 9th received data to the left of the MSB of the data register, R8 needs to be read out before reading the data register, because reading the data register can complete the automatic receive full flag clearing, allowing R8 and SCI_D to be overwritten by new data.
T8	The 9th data of the transmitter. When SCI is configured for 9-bit data, T8 can be considered the 9th transmit data bit to the left of the MSB that buffers the data in the data register. T8 should be written before the data register is written (if it needs to be modified from its previous value). If T8 does not need to be modified from its new value (e.g. when it is used to generate a flag or parity), it does not need to be written before each write to the data register. When parity enable is configured, this bit is used as a parity bit and is automatically generated by hardware, not a written value.
The TXD_DIRECT	The TXD_DIRECT bit is meaningful when CYCLE_MODE = 1 and SINGLE_TXD = 1, and is used to select the txd pin direction.
TXD_INV	Invert the data at the txd, it will invert all output signals. The interrupt enable register, which is set to 1 to enable interrupt generation and is set to 0 to

	disable interrupt generation, but does not affect the interrupt flag set in the following register, used for polling mode.
RXD_INV	Invert the data at the rxd IO, it will invert all the input signals.
RWU_IDLESEL	Receive wakeup from idle detection, used to control whether the IDLE state and interrupts are set during receiver standby.
TXD_DIRECT/TXD_INV/RXD_INV/RWU_IDLESEL/IDLE_SEL/WAKE_SEL	TXD_DIRECT/TXD_INV/RXD_INV/RWU_IDLESEL/IDLE_SEL/WAKE_SEL registers are also valid in LIN frame mode.

14.10.7 Sync Segment Control Register (SCI_S2)

Address: 0xBA

Bit No.	7	6	5	4
Symbol	BREAK_CHECK_IF	RX_EDGE_IF	RX_ACTIVE_FLAG	RXDATA0_FRAME_CHECK
Read/Write	Read/Write	Read/Write	Read	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	BREAK_CHECK_IE	RX_EDGE_IE	BREAK_TRANS_SIZE	BREAK_CHECK_EN
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	BREAK_CHECK_IF	Interval segment detection interrupt flag. 1: The interval segment is detected. 0: No interval segment is detected, the bit is cleared by writing 1 and invalid by writing 0
6	RX_EDGE_IF	RxD pin active edge interrupt flag. 1: Active edge occurs on the receive pin. 0: No active edge on the receive pin, this bit is cleared by writing 1, invalid by writing 0
5	RX_ACTIVE_FLAG	Receiver activity flag, read-only. 1: Receiver active 0: Receiver idle

4	RXDATA0_FRAME_CHECK	Interval detection enable bit, configuring delayed transmit interrupt when 0x00 character is detected and STOP=0 while interval detection interrupt is disabled 1: Transmit interrupt received and frame error interrupt delayed. 0: Ignore this byte data reception
3	BREAK_CHECK_IE	Interval segment detection interrupt enable. 1: Interrupt enabled 0: Interrupt disabled
2	RX_EDGE_IE	RxD pin active edge interrupt enable. 1: Interrupt enabled 0: Interrupt disabled
1	BREAK_TRANS_SIZE	The length of the interval segment generation bit. 1: transmit with 13-bit times length (1 bit more if DATA_MODE = 1 or STOP_MODE = 1, respectively) 0: transmit with 10-bit times length (if DATA_MODE = 1 or STOP_MODE = 1, add 1 bit time respectively)
0	BREAK_CHECK_EN	Interval detection enable. 1: Detect at 11-bit time length (if DATA_MODE = 1 or STOP_MODE = 1, add 1 bit time, respectively) 0: Not detected

Symbol	Description
BREAK_CHECK_IF	Interval segment detection interrupt flag, which is set when BREAK_CHECK_EN=1 and the LIN abort character is detected. --The characters of bit 11 /12/13 are detected as 0. To clear this flag: The software writes 1 to this register bit directly.
RX_EDGE_IF	Active edge interrupt flag on RxD pin, which is set when there is an active edge on the RxD pin (falling if RXD_INV = 0, rising if RXD_INV = 1). To clear this flag: The software writes a 1 to this register bit directly.
RX_ACTIVE_FLAG	Receiver activity flag that is set when the SCI receiver detects the start of a valid start bit and cleared when the receiver detects an idle line. This status flag can be used to tell the MCU if it is currently receiving SCI characters. This bit is read-only.
	Interval segment detection enable bit. When 0x00 character is detected and STOP=0 and interval

RXDATA0_FRAME_CHECK	segment detection interrupt is failed, configure delayed transmit interrupt. 1 means delayed transmit receive interrupt and frame error interrupt while 0 means to ignore this data reception. When 10 consecutive "0" bits are received and the 11th bit of the interval detection is "1", the frame error flag is allowed to delay the transmission of this data.
BREAK_CHECK_IE	interval segment detection interrupt enable, used to control the generation of interrupt signal.
RX_EDGE_IE	RxD pin active edge interrupt enable, used to control the generation of interrupt signal.
BREAK_TRANS_SIZE	The size of the interval generation bits, used to select whether to send a 13-bit interval.
BREAK_CHECK_EN	Interval segment detection circuit enable. When this bit is set, if the STOP bit after the detection of 0x00 characters still detects 0 characters, then it will prevent the frame error and receive data register full flag is set and continue to detect the next 0 characters.

BREAK_TRANS_SIZE	DATA_MODE	STOP_MODE	Character length
0	0	0	10 bit times
0	0	1	11 bit times
0	1	0	11 bit times
0	1	1	12 bit times
1	0	0	13 bit times
1	0	1	14 bit times
1	1	0	14 bit times
1	1	1	15 bit times

Table 14 .10 Interval segment lengths

Note:

If normal data is detected after enabling interval segment detection, it will be received normally, including frame error and receive full will be set normally. Once 0x00 character is detected and

STOP=0, the interval segment will be detected first, and if the next bit is still 0, it will be determined as interval segment. If the next bit detects a rising edge, you can choose whether to delay transmitting the receive interrupt and frame error interrupt via the register RXDATA0_FRAME_CHECK. If this bit is not enabled, all data is received normally and frame error and receive full are set normally.

This register is also valid in LIN frame mode, where after BREAK_CHECK_EN is set, it will always detect the BREAK segment.

14.10.8 Interrupt Status Flag Register (SCI_S1)

Address: 0xBB

Bit No.	7	6	5	4
Symbol	TX_EMPTY_IF	TX_FINISH_IF	RX_FULL_IF	IDLE_IF
Read/Write	Read	Read	Read	Read
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	RX_OVERFLOW_IF	NOISE_ERR_IF	FRAME_ERR_IF	PARITY_ERR_IF
Read/Write	Read	Read	Read	Read
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	TX_EMPTY_IF	Transmit a cache empty interrupt flag. 1: Transmit cache is empty, 0: Transmit cache is full, read-only
6	TX_FINISH_IF	Transmit the finish interrupt flag. 1: Transmit is completed and the transmitter is idle 0: Transmitter is working, read only
5	RX_FULL_IF	Receive a full interrupt flag. 1: The receive cache is full. 0: The receive cache is empty, read-only
4	IDLE_IF	idle line interrupt flag. 1: Idle line detected. 0: No idle line detected, read-only
3	RX_OVERFLOW_IF	Receive the overflow interrupt flag. 1: Received overflow (new data lost). 0: No overflow, read-only
2	NOISE_ERR_IF	noise interrupt flag. 1: Noise detected. 0: No noise detected, read-only

1	FRAME_ERR_IF	frame error interrupt flag. 1: Frame error detected 0: No frame error detected, read-only
0	PARITY_ERR_IF	Parity error interrupt flag. 1: Receiver parity error 0: Parity correct, read-only

Symbol	Description
TX_EMPTY_IF	The transmit buffer empty interrupt flag, which is set when the transmit data value is transferred from the transmit data buffer (tx_buffer) to the transmit shifter to make space in the buffer for the new character. The interrupt flag can be cleared by writing to the SCI data register (SCI_D) and writing to the corresponding clear register CLK_TX_EMPTY_IF.
TX_FINISH_IF	Transmit finished interrupt flag, which is set when the transmitter has finished sending all data, idle characters and abort characters, and it is in the idle state. The methods to clear this interrupt flag are: writing to SCI data register (SCI_D) to send new data; queuing idle characters by changing TRANS_ENABLE from 0 to 1; queuing abort characters by writing BREAK_TRANS_START to 1 and then clearing it to 0, and writing the corresponding clear register CLK_TX_FINISH_IF.
RX_FULL_IF	Receive full interrupt flag, which is set when the character is transferred from the receive shifter to the receive data register (SCI_D). The interrupt flag can be cleared by reading out the SCI data register and writing the corresponding clear register CLK_RX_FULL_IF.
	Idle line interrupt flag that is set IDLE after a period of activity when the SCI receive line has been idle for a full character time. When IDLE_SEL=0, the receiver starts counting idle bit times after the start bit. Therefore, if the receive characters are all 1s, these bit times and the stop bit times count into the

IDLE_IF	full character time that the receiver uses to detect the required logical high state (10/11/12 bit times, depending on the DATA_MODE and STOP_MODE control bits) for an idle line. When IDLE_SEL=1, the receiver does not start counting the idle bit time until after the stop bit. Therefore, the stop bit and any logical high state bit time at the end of the previous character will not count into the full character time of the logical high state that the receiver uses to detect an idle line. The interrupt flag is cleared by reading the SCI data register and writing the corresponding clear register CLK_IDLE_IF. After clearing this flag, it will not be set again until a new character is received and RX_FULL_IF has been set.
RX_OVERFLOW_IF	Receive overflow interrupt flag, which is set when a new serial character is received to be transferred to the receive data register and the original received character has not been read from SCI_D. In this case, the new character is lost (all associated error flags are lost). The interrupt flag can be cleared by reading the SCI data register and writing the corresponding clear register CLK_RX_OVERFLOW_IF.
NOISE_ERR_IF	Noise interrupt flag, which is set when the receiver samples 16 samples per bit, if the samples are inconsistent then it is flagged (8 samples from RT3 to RT10 for the start bit, 3 samples from RT8 to RT10 for the other bits, including the stop bit). The interrupt flag can be cleared by reading out the SCI data register and writing the corresponding clear register CLK_NOISE_ERR_IF.
FRAME_ERR_IF	Frame error interrupt flag, which is set when the receiver detects a logic 0 when it should be a stop bit, and the receive full interrupt flag is also set. The interrupt flag can be cleared by reading out the SCI data register and writing the corresponding clear register CLK_FRAME_ERR_IF.
	Parity error interrupt flag, which is set when parity is enabled (PARITY_EN=1) and the parity bit in the

PARITY_ERR_IF	received character does not match the expected parity value. To clear this interrupt flag, read out the SCI data register and write the corresponding clear register CLK_PARITY_ERR_IF.

The above three error interrupt flags only record the error of the data currently deposited in the data register, and the data update status is automatically updated.
This register is not valid in LIN frame mode.

14.10.9 SCI Data Register (SCI_D)

Address: 0xBC

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read/Write							
Po Initial Value	FF							

Bit No.	Symbol	Description
7~0	--	SCI Data Register Read returns the contents of the read-only receiving data buffer, write goes to the write-only transmitting data buffer.

One set of addresses corresponds to two sets of cache registers. Reads return the contents of the read-only receiving data buffer and write goes to the write-only transmitting data buffer.
This register is not valid in LIN frame mode

14.10.10 SCI Interrupt Clear Register (SCI_INT_CLR)

Address: 0xBD

Bit No.	7	6	5	4
Symbol	CLK_TX_EMPTY_IF	CLK_TX_FINISH_IF	CLK_RX_FULL_IF	CLK_IDLE_IF
Read/Write	Write	Write	Write	Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0

Symbol	CLK_RX_OVERFLOW_IF	CLK_NOISE_ERR_IF	CLK_FRAME_ERR_IF	CLK_PARITY_ERR_IF
Read/Write	Write	Write	Write	Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	CLK_TX_EMPTY_IF	Transmit buffer empty interrupt clear bit. Writing 1 to this bit clears the corresponding interrupt, while write 0 is invalid.
6	CLK_TX_FINISH_IF	Transmit finish interrupt clear bit. Writing 1 to this bit clears the corresponding interrupt, while write 0 is invalid.
5	CLK_RX_FULL_IF	Receive full interrupt clear bit. Writing 1 to this bit clears the corresponding interrupt, while write 0 is invalid.
4	CLK_IDLE_IF	Idle line interrupt clear bit. Writing 1 to this bit clears the corresponding interrupt, while write 0 is invalid.
3	CLK_RX_OVERFLOW_IF	Receive overflow interrupt clear bit. Writing 1 to this bit clears the corresponding interrupt, while write 0 is invalid.
2	CLK_NOISE_ERR_IF	Noise interrupt clear bit. Writing 1 to this bit clears the corresponding interrupt, while write 0 is invalid.
1	CLK_FRAME_ERR_IF	Frame error interrupt clear bit. Writing 1 to this bit clears the corresponding interrupt, while write 0 is invalid.
0	CLK_PARITY_ERR_IF	Parity error interrupt clear bit. Writing 1 to this bit clears the corresponding interrupt, while write 0 is invalid.

This register can only be written, not read. Write 1 to corresponding bit to clear the corresponding interrupt flag, while writing 0 is invalid. This register is only used as a backup and does not need to be configured for general applications.
This register is not valid in LIN frame mode.

14.10.11 LIN Frame Timeout Configuration Register (LIN_FRAME_TIMEOUT)

XRAM_SFR Address: 0x2080

Bit No.	7	6	5	4	3	2	1	0
Symbol	LIN_FRAME_TIMEOUT							
Read/Write	Read/Write							
Po Initial Value	0	0	0	0	0	0	0	0

Bit No.	Symbol	Description
7~0	--	Frame timeout configuration. It requires to select the corresponding configuration according to the frame timeout mode and the number of data fields

This register is used for frame timeout comparison, and the corresponding configuration is selected according to the mode of frame timeout and the number of data fields. The minimum is 34*1.4, and the maximum 124*1.4.

14.10.12 LIN Configuration Register 1 (LIN_CTL1)

XRAM_SFR Address: 0x2081

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	-	-	BIT_ERROR_IGNORE	FRAME_MODE
Read/Write	-	-	-	-	-	-	Read/Write	Read/Write
Po Initial Value	-	-	-	-	-	-	0	0

Bit No.	Symbol	Description
1	BIT_ERROR_IGNORE	Transmission mode selection upon bit error. 0: Terminate transmission, 1: Continue transmission
0	FRAME_MODE	LIN frame mode enable. 0: Disabled, 1: Enabled

Symbol	Description
BIT_ERROR_IGNORE	Selects whether the transmission needs to be terminated after a bit error is detected.
FRAME_MODE	LIN frame mode enable selection. When enabled, UART mode is automatically disabled, only communicating according to the frame command format in order.

14.10.13 LIN Configuration Register 0 (LIN_CTL0)

XRAM_SFR Address: 0x2082

Bit No.	7	6	5	4
Symbol	FRAME_TIMEOUT_SEL		BREAK_DELIMITER_LEN	
Read/Write	Read/Write		Read/Write	
Po Initial Value	0		0	
Bit No.	3	2	1	0
Symbol	CHECKSUM_ENHANCED	DATA_NR		
Read/Write	Read/Write	Read/Write		
Po Initial Value	0	0		

Bit No.	Symbol	Description
7~6	FRAME_TIMEOUT_SEL	Mode selection for frame timeout 0: Frame timeout not enabled; 1: Frame timing. 2: Frame header timing; 3: Frame response timing
5~4	BREAK_DELIMITER_LEN	Interval configuration of Frame synchronization interval segment. 0~3: 2~5 interval bits
3	CHECKSUM_ENHANCED	Checksum mode selection. Standard checksum does not contain PID, enhanced checksum contains PID, the protocol specifies that the standard checksum (ID=0x3c/0x3d) is fixed for diagnostic frames. Here the hardware determines it automatically, independent of this register configuration. 0: Standard checksum; 1: Enhanced checksum
2~0	DATA_NR	LIN frame data segment length, 0~7: 1 to 8 bytes The number of data that needs to be transmitted and received uniformly by the application side before transmission starts.

Symbol	Description
FRAME_TIMEOUT_SEL	Select the timing mode for the frame timeout.
BREAK_DELIMITER_LEN	Select the number of interval bits for the frame synchronization interval. 0~3: 2~5 interval bits, the default is 2 interval bits.
CHECKSUM_ENHANCED	Checksum mode selection, standard checksum does not contain PID, enhanced checksum contains PID, the protocol specifies that the standard checksum (ID=0x3c/0x3d) is fixed for diagnostic frames. Here the hardware automatically determines, independent of this register configuration. 0: standard checksum; 1: enhanced

	checksum. The way to calculate check sum is to add each byte of the checksum object in rounded binary (subtract 255 whenever the result is greater than or equal to 256), and invert the final sum bit by bit, and use the result as the checksum to be sent. The receiver operates the same rounded binary addition to the received data according to the checksum type, without inverting the final sum, and adds the sum to the received checksum, and if the result is 0xFF, the checksum is correct.
DATA_NR	length of LIN frame data segment, 0~7: 1 to 8 bytes.

14.10.14 LIN Transfer Command Register (LIN_LIN_CMD)

XRAM_SFR Address: 0x2083

Bit No.	7	6	5	4
Symbol	-			
Read/Write	-			
Po Initial Value	-			
Bit No.	3	2	1	0
Symbol	RX_RESPONSE	RX_HEADER	TX_RESPONSE	TX_HEADER
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
3	RX_RESPONSE	Receive frame response
2	RX_HEADER	Receive frame header
1	TX_RESPONSE	Sending frame response
0	TX_HEADER	Sending frame header

Symbol	Description
	Software configures 1 to start receiving response, after the STOP bit of the checksum is received to

RX_RESPONSE	determine the response reception is complete, any valid transmission is completed (one complete frame configuration) will have hardware to clear, if there is an error not cleared.
RX_HEADER	Software configures 1 to start receiving frame headers, determines frame header reception completion after the STOP bit of the PID is received, any valid transmission completion (one complete frame configuration) will have hardware to clear, if there is an error it is not cleared. Usually, the slave node has to set two instructions RX_HEADER and RX_RESPONSE, because the master node is expected to transmit the response to the slave node. After receiving the PID, the slave node can set TX_RESPONSE, which has a higher priority. BREAK is always in detection, with or without a receive frame header command, and after BREAK is detected (BREAK_CHECK_IF), the hardware checks the receive frame header command before entering the synchronization segment, and when the command is 0, the software has at least 11 bits to reset the receive frame header command before the next BREAK (BREAK_CHECK_IF). This way, BREAK is not lost.
TX_RESPONSE	Software configures 1 to start transmitting response, determine transmission completion after the STOP bit of the transmit checksum is completed, any valid transmission completion (one complete frame configuration) will have hardware to clear. If there is an error, it will immediately clear to zero.
TX_HEADER	Software configures 1 to start sending frame headers, determine transmission completion after sending STOP bit of PID, any valid transmission completion (one complete frame configuration) will be hardware cleared, if there is an error, it will immediately clear to 0.

14.10.15 LIN Synchronization Section Baud Rate Count Period Register (LIN_SYNC_CO UNTER)

XRAM_SFR Address: 0x2084

Bit No.	7	6	5	4	3	2	1	0
Symbol	-							
Read/Write	Read							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	--	Read-only, used to record the baud rate counting period of the synchronous segment, if there is no deviation, it should be 8*16=128. The deviation allowed in the frame protocol is 0.5% at the transmitter side and 14% at the receiver side, and the deviation allowed in the synchronous segment is within [106, 152] according to this calculation.

14.10.16 LIN Status Register (LIN_STATUS)

XRAM_SFR Address: 0x2085

Bit No.	7	6	5	4
Symbol	TX_HEADER_BREAK_DELIMITER_ERROR	TX_HEADER_BREAK_ERROR	RX_DONE	TX_DONE
Read/Write	Read	Read	Read	Read
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	RX_BUSY	TX_BUSY	RX_DATA0_FRAME_ERROR	HEADER_RESPONSE
Read/Write	Read	Read	Read	Read
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	TX_HEADER_BREAK_DELIMITER_ERROR	A "0" is detected when the host sends the break delimiter of the frame header and is cleared when the current instruction is completed.
6	TX_HEADER_BRE	A "1" is detected when the host sends the BREAK of the frame header

	AK_ERROR	and is cleared when the current instruction is completed.
5	RX_DONE	Receive done bit, set after received the instruction and cleared to 0 upon a new command. - RX_HEADER, RX_RESPONSE. - TX_HEADER, RX_RESPONSE.
4	TX_DONE	Transmit done bit, set after transmit the instruction and cleared to 0 upon a new command. - TX_HEADER. - TX_HEADER, TX_RESPONSE. - RX_HEADER, TX_RESPONSE.
3	RX_BUSY	Receiver in busy state, set at the beginning of the receive frame header and frame response, cleared when the current command is completed or there is an error.
2	TX_BUSY	Transmitter busy state, set at the beginning of the transmitting frame header and frame response, cleared at the end of the current command or if there is an error.
1	RX_DATA0_FRAME_ERROR	The frame error flag for the first data in the frame response, which is set when the first data received is 0x00 and the STOP bit detects 0 (only after receiving the frame header command), it will clear when the falling edge of the START bit of the synchronization segment of the BREAK segment is finished receiving. This bit, together with the receive response frame error interrupt flag, distinguishes whether there is no response or an error response or a correct response is received. This flag does not abort the message transmission.When it is set, the description is that this frame does not response and needs no handling.RX_RESPONSE_FRAME_ERROR is set, Description is an error and needs to be processed, Description is a correct response when neither is set.
0	HEADER_RESPONSE	The frame header response status, set at the beginning of the frame response and cleared at the end of the frame response. To be combined with TX_BUSY=1 or RX_BUSY=1. 0: Frame header transmission in progress. 1: Frame response in transmission.

14.10.17 LIN Data Field Count Value Register (LIN_DATA_IDX)

XRAM_SFR Address: 0x2086

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-	-	--			
Read/Write	-	-	-	-	Read			

Po Initial Value	-	-	-	-	0

Bit No.	Symbol	Description
3~0	--	Data field count value, which is cleared to 0 at the beginning of the transmit/receive frame header, indicating which byte number of data currently being transmitted. 0: No bytes transferred. 1: 1 byte Transferred. ... 7: 1~7 bytes transferred. 8: 1~8 bytes transferred. 9: 1~8 bytes and checksum bytes transferred. 10~15: Unused.

14.10.18 LIN Interrupt Status Register 1 (LIN_INTR1)

XRAM_SFR Address: 0x2087

Bit No.	7	6	5	4
Symbol	RX_RESPONSE_CHECKSUM_ERROR	RX_RESPONSE_FRAME_ERROR	RX_HEADER_PARITY_ERROR	RX_HEADER_SYNC_ERROR
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	RX_HEADER_FRAME_ERROR	TX_RESPONSE_BIT_ERROR	TX_HEADER_BIT_ERROR	TIMEOUT
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	RX_RESPONSE_CHECKSUM_ERROR	Hardware sets the software to write 1 to clear, while writing 0 has no effect. Receive response checksum error, which will be reset when the calculated checksum does not match the received one. Ongoing info transmission is aborted and the transmit command is set to 0.
6	RX_RESPONSE_FRAME_ERROR	Hardware sets the software to write 1 to clear, while writing 0 has no effect. Receive response frame error, which will be reset when there is abnormal data in the received START/STOP. If a 0 is received, this bit is not used to determine the stop bit for the first data after the RX_HEADER command,and instead the RX_DATA0_FRAME_ERROR is used. Ongoing info transmission is aborted and the transmit command is set to 0
5	RX_HEADER_PARITY_ERROR	Hardware sets the software to write 1 to clear, while writing 0 has no effect. Receive frame header PID checksum error, which is performed automatically in LIN frame mode, independent of the parity enable bit in uart mode. Ongoing info transmission is aborted and the transmit command is set to 0
4	RX_HEADER_SYNC_ERROR	Hardware sets the software to write 1 to clear, while writing 0 has no effect.

		Receive frame header synchronous segment error, set if the syncing segment is not received within the specified time (SYNC_COUNTER = [106, 152]). The ongoing info transmission is aborted and the transmit command is set to 0
3	RX_HEADER_FRAME_ERROR	Hardware sets the software to write 1 to clear, while writing 0 has no effect. Receive frame header frame error, which is set when there is abnormal data in the received START/STOP. Ongoing info transmission is aborted and the transmit command is set to 0
2	TX_RESPONSE_BIT_ERROR	Hardware sets the software to write 1 to clear, while writing 0 has no effect. Send response bit error, set when the levels of transmit line and receive line do not match. It is used to transmit all positions of the response data byte and checksum (START,DATA,STOP). When the bit error ignore function (BIT_ERROR_IGNORE=1) is configured, this error does not cause the info transmission to be aborted, otherwise the ongoing info transmission is aborted and the send command is set to 0
1	TX_HEADER_BIT_ERROR	Hardware sets the software to write 1 to clear, while writing 0 has no effect. Transmit frame header bit error, set when transmit line and receive line levels do not match. It is used to send all positions of the interval segment, sync byte and PID of the frame header (START,DATA,STOP). When the bit error ignore function (BIT_ERROR_IGNORE=1) is configured, this error does not cause the info transmission to be aborted, otherwise the ongoing info transmission is aborted and the send command is set to 0.
0	TIMEOUT	Hardware sets the software to write 1 to clear, while writing 0 has no effect. Timeout error, three configuration timeout calculation methods, set after timeout.

14.10.19 LIN Interrupt Status Register 2 (LIN_INTR2)

XRAM_SFR Address: 0x2088

Bit No.	7	6	5	4
Symbol	-			RX_HEADER_SYNC_DONE
Read/Write	-			Read/Write

Po Initial Value	-			0
Bit No.	3	2	1	0
Symbol	RX_RESPONSE_DONE	RX_HEADER_DONE	TX_RESPONSE_DONE	TX_HEADER_DONE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
4	RX_HEADER_SYNC_DONE	Hardware sets the software to write 1 to clear, while writing 0 has no effect. The synchronous segment of Frame received header completion flag. Set after receiving the STOP bit.
3	RX_RESPONSE_DONE	Hardware set software to write 1 to clear, while writing 0 has no effect. Receive frame response completion flag. This bit can only be set if no receive response error is triggered
2	RX_HEADER_DONE	Hardware sets software to write 1 clear, while writing 0 has no effect. Receive frame header completion flag. Set after the full frame header has been received.
1	TX_RESPONSE_DONE	Hardware sets software to write 1 clear, while writing 0 has no effect. Transmit frame response completion flag.
0	TX_HEADER_DONE	Hardware sets software to write 1 clear, while writing 0 has no effect. Transmit frame header completion flag.

14.10.20 LIN Interrupt Enable Register 1 (LIN_INTR1_IE)

XRAM_SFR Address: 0x2089

Bit No.	7	6	5	4
Symbol	RX_RESPONSE_CHECKSUM_ERROR_IE	TRX_RESPONSE_FRAME_ERROR_IE	RX_HEADER_PARITY_ERROR_IE	RX_HEADER_SYNC_ERROR_IE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0
Bit No.	3	2	1	0
Symbol	RX_HEADER_FRAME_ERROR_IE	TX_RESPONSE_BIT_ERROR_IE	TX_HEADER_BIT_ERROR_IE	TIMEOUT_IE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
7	RX_RESPONSE_CHECKSUM_ERROR_IE	Receive response checksum error interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
6	RX_RESPONSE_FRAME_ERROR_IE	Response frame received error interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
5	RX_HEADER_PARITY_ERROR_IE	Receive header PID error interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
4	RX_HEADER_SYNC_ERROR_IE	Synchronous segment error interrupt of Frame received header enable. 1: Interrupt enabled. 0: Interrupt disabled.
3	RX_HEADER_FRAME_ERROR_IE	Frame received header error interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
2	TX_RESPONSE_BIT_ERROR_IE	Transmit response bit error interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
1	TX_HEADER_BIT_ERROR_IE	Transmit frame header bit error interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
0	TIMEOUT_IE	Timeout error interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.

14.10.21 LIN Interrupt Enable Register 2 (LIN_INTR2_IE)

XRAM_SFR Address: 0x208A

Bit No.	7	6	5	4
Symbol	-			RX_HEADER_SYNC_DONE_IE
Read/Write	-			Read/Write
Po Initial Value	-			0
Bit No.	3	2	1	0
Symbol	RX_RESPONSE_DO	RX_HEADER_DONE_IE	TX_RESPON	TX_HEADE

	NE_IE		SE_DONE_IE	R_DONE_IE
Read/Write	Read/Write	Read/Write	Read/Write	Read/Write
Po Initial Value	0	0	0	0

Bit No.	Symbol	Description
4	RX_HEADER_SYNC_DONE_IE	Frame received header to synchronously interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
3	RX_RESPONSE_DONE_IE	Response received to interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
2	RX_HEADER_DONE_IE	Frame Received header to interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
1	TX_RESPONSE_DONE_IE	Frame response transmitted to interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.
0	TX_HEADER_DONE_IE	Frame transmitted header to interrupt enable. 1: Interrupt enabled. 0: Interrupt disabled.

14.10.22 LIN Transmit Register Description

When configuring the transmit data register, it needs to be done before configuring the transmit command, and no further configuration is allowed during transmission.

14.10.22.1 LIN Transmit PID Register (LIN_TX_PID)

XRAM_SFR Address: 0x2090

Bit No.	7	6	5	4	3	2	1	0
Symbol	-	-	-					
Read/Write	-	-	Read/Write					
Po Initial Value	-	-	0					

Bit No.	Symbol	Description
5~0	--	Store PIDs transmitted in LIN frame mode When transmitting, It is calculated automatically according to the written bit 0 to 5 as follows:

		bit6:P0 = ID0⊕ ID1⊕ ID2⊕ ID4 bit7:P1 = ~ (ID1⊕ ID3⊕ ID4⊕ ID5)

14.10.22.2 LIN Transmit Data Register (LIN_TX_DATA)

XRAM_SFR Address: 0x2091~0x2098

Bit No.	7	6	5	4	3	2	1	0
Symbol	LIN_TX_DATAx							
Read/Write	Read/Write							
Po Initial Value	0							

LIN_TX_DATAx, where x: 1~8

Bit No.	Symbol	Description
7~0	LIN_TX_DATAx	Stores the transmitted data field and byte X in LIN frame mode, where x ranges from 1 to 8.

14.10.22.3 LIN Transmit Checksum Register (LIN_TX_ CHECKSUM)

XRAM_SFR Address: 0x2099

Bit No.	7	6	5	4	3	2	1	0
Symbol	LIN_TX_CHECKSUM							
Read/Write	Read/Write							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	LIN_TX_CHECKSUM	Store the transmitted checksum in LIN frame mode

14.10.23 LIN Receive Register Description

Received data is not lost after the SCI module is disabled.

14.10.23.1 LIN Receive PID Register (LIN_RX_PID)

XRAM_SFR Address: 0x20A0

Bit No.	7	6	5	4	3	2	1	0
Symbol	LIN_RX_PID							
Read/Write	Read							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	LIN_RX_PID	Store the received PID in LIN frame mode

14.10.23.2 LIN Receive Data Register (LIN_RX_DATA)

XRAM_SFR Address: 0x20A1~0x20A8

Bit No.	7	6	5	4	3	2	1	0
Symbol	LIN_RX_DATAx							
Read/Write	Read							
Po Initial Value	0							

LIN_RX_DATAx, where x: 1 to 8.

Bit No.	Symbol	Description
7~0	LIN_RX_DATAx	Stores the received data field and byte x in LIN frame mode, where x ranges from 1 to 8.

14.10.23.3 LIN Receive Checksum Register (LIN_RX_CHECKSUM)

XRAM_SFR Address: 0x20A9

Bit No.	7	6	5	4	3	2	1	0
Symbol	LIN_RX_CHECKSUM							
Read/Write	Read							
Po Initial Value	0							

Bit No.	Symbol	Description
7~0	LIN_RX_CHECKSUM	Store the received checksum in LIN frame mode

15 Programming and Debugging

FLASH uses a two-wire serial command Programming method. In addition to the power and ground, 2 IOs are used for the Programming clock line (PGC) and data line (PGD).

15.1 SWE Circuit Connection

Two-wire Programming and single-wire debugging. During the emulation debugging, you need to connect one wire of SWE. In the SWE debugging mode, the IO function of the SWE is masked, it is recommended not to configure other functions of the SWE debugging IO port to avoid affecting the SWE debugging function.

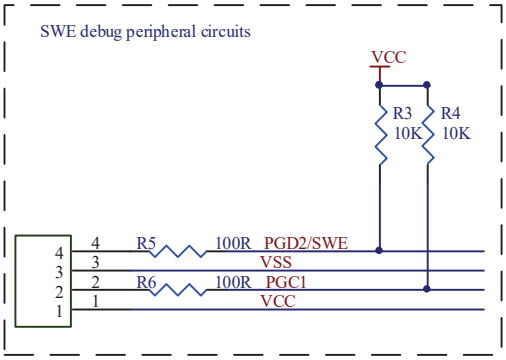
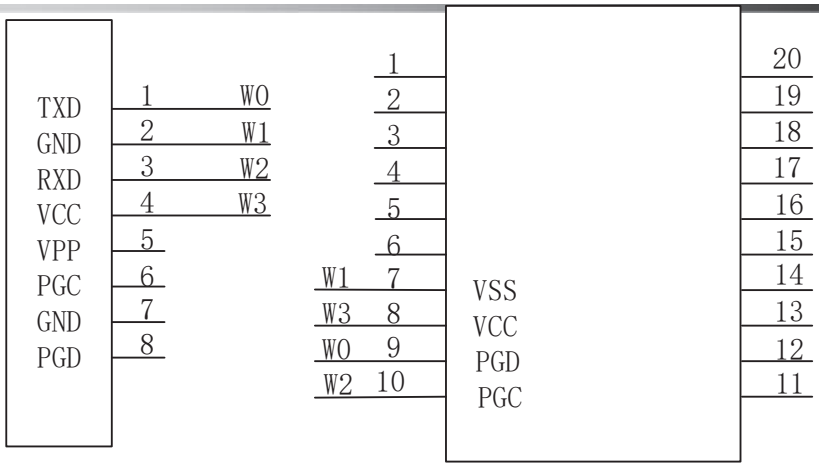


Table 15 .1 SWE circuit connection reference diagram

15.2 Capacitive TouchKey Data Assisted Programming & debugging

Connect the four wires of chip PGC (PGC1/PGC2), PGD (PGD1/PGD2), VCC and VSS, and when you enter the Programming interface, select the corresponding model of chip, open the compiled HEX file, click on the one-click write FLASH and wait for the Programming to complete.
When you enter the debugging interface, first burn the HEX file with debugging data sending mode, and click Start Debug to view the button data.
Note: Refer to the Capacitive button Programming/debugging guide for specific operation descriptions.



ConnectPinsMin

Table 0.1 BS9000AMxx Programming Wiring Diagram

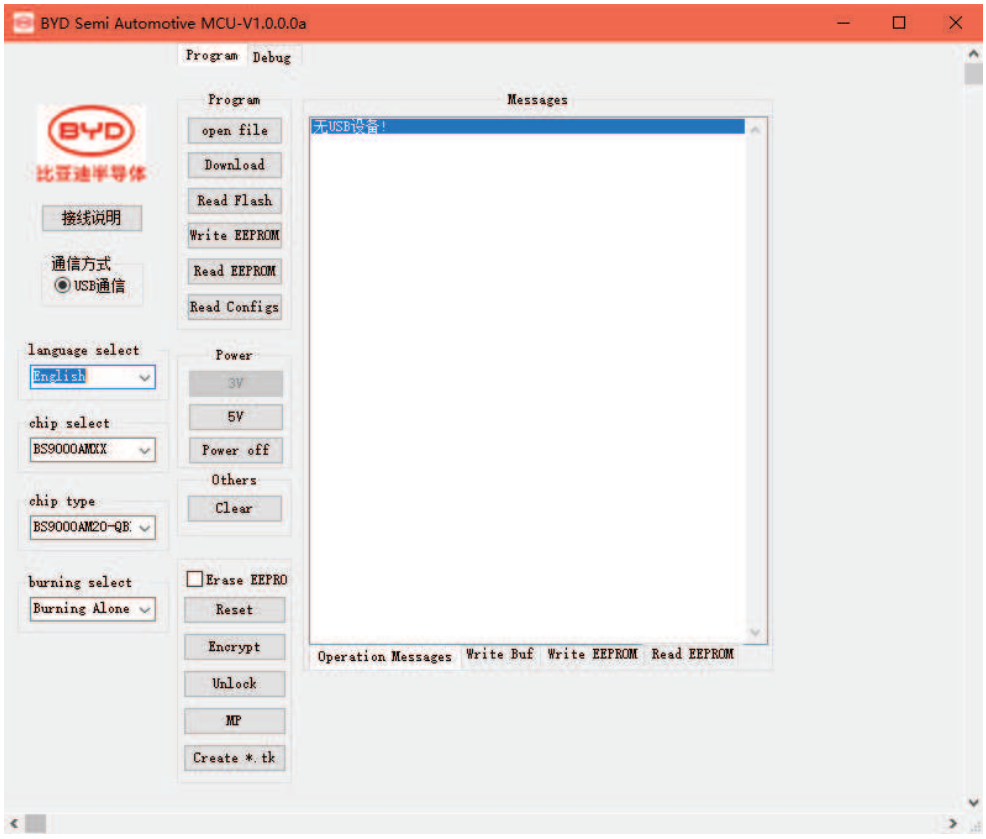


Table 15 .3 Programming Interface

16 CPU Instructional System

16.1 Instruction Code

The instructions of BS9000AMxx are divided into 1-byte instruction, 2-byte instructions and 3-byte instructions.

1-byte instruction: 1-byte instruction consists of an 8-bit binary code. There is only instruction opcode in the instruction, no instruction operand or instruction operand is implied in the instruction opcode. There are 49 instructions in this category.

2-byte instructions: 2-byte instructions consist of two bytes, one for the opcode and the other for the operand (or the address of the operand), which are stored sequentially in program memory. There are 46 instructions in this class.

3-byte instructions: 3-byte instructions consist of a one-byte instruction opcode and a two-byte operand (or the address of the operand). There are 16 instructions in this class.

16.2 Instruction Set

In order to describe the instructions more clearer and easier, some Symbols are used in the instructions, of which meanings are as follows.

addr 11	Low 11-bit address
addr 16	16-bit address
direct	Direct addressable, 8-bit internal data and address (including SFRs)
bit	Bit Address
#data	8-bit immediate number
#data16	16-bit immediate number
rel	8-bit relative displacement with Symbol
n	digit 0 to 7
Rn	R0 to R7 working registers of current register set
i	Digit 0, 1
Ri	Working registers R0, R1
@	Register indirect addressing
←	Data transfer direction
∧	Logic "with"
∨	Logic "or"
⊕	Logic "Xor"
√	Has an effect on the flag bits
×	No effect on flag bits

Table 16 .1 CPU Instruction Symbol

The table 16.2 shows assembly instructions, function of each instruction, the number of bytes occupied, the instruction execution cycle, and the effect on the corresponding flag bits that provided for use.

8-bit Data Transfer Instructions								
Mnemonic sign		Function	Effect on flag bits				Number of bytes	Number of cycles
			P	OV	AC	CY		
MOV A	Rn	A←(Rn)	√	×	×	×	1	1
	direct	A←(direct)	√	×	×	×	2	1
	@Ri	A←((Ri))	√	×	×	×	1	1
	#data	A←data	√	×	×	×	2	1
MOV Rn	A	Rn←(A)	×	×	×	×	1	1
	direct	Rn←(direct)	×	×	×	×	2	2
	#data	Rn←data	×	×	×	×	2	1
MOV direct1	A	direct1←(A)	×	×	×	×	2	1
	Rn	direct1←(Rn)	×	×	×	×	2	2
	direct2	direct1←(direct2)	×	×	×	×	3	2
MOV direct	@Ri	direct←((Ri))	×	×	×	×	2	2
	#data	direct←data	×	×	×	×	3	2
MOV @Ri	A	(Ri)←(A)	×	×	×	×	1	1
	direct	(Ri)←(direct)	×	×	×	×	2	2
	#data	(Ri)←data	×	×	×	×	2	1
16-bit data transfer instruction								
mnemonic sign		Function	Effect on flag bits				Number of bytes	Number of cycles
			P	OV	AC	CY		
MOV DPTR,#data16		DPTR←data16		×	×	×	3	2
External data transfer and table lookup instructions								
mnemonic sign		Function	Effect on flag bits				Number of bytes	Number of cycles
			P	OV	AC	CY		
MOVX @DPTR,A		(DPTR)←(A)	×	×	×	×	1	2
MOVC A,	@A+DPTR	A←((A)+(DPTR))	√	×	×	×	1	2
	@A+PC	A←((A)+(PC))	√	×	×	×	1	2
MOVX A,	@DPTR	A←(DPTR)	√	×	×	×	1	2
Note: The number of cycles and the number of bytes of the MOVX instruction can be configured via register CKCON<2:0>.								
Exchange Instructions								
mnemonic sign		Function	Effect on flag bits				Number of bytes	Number of cycles
			P	OV	AC	CY		

XCH A,	Rn	(Rn)←(A)	√	×	×	×	1	2
	direct	(A)←(direct)	√	×	×	×	2	1
	@Ri	(A)←((Ri))	×	×	×	×	1	1
XCHD A,@Ri		(A)3~0~((Ri))3~0	√	×	×	×	1	1
SWAP A		(A)7-4~(A)3-0	√	×	×	×	1	1
Arithmetic operation instruction								
mnemonic sign		Function	Effect on flag bits				Number of bytes	Number of cycles
			P	OV	AC	CY		
ADD A,	Rn	A←(A)+(Rn)	√	√	√	√	1	1
	direct	A←(A)+(direct)	√	√	√	√	2	1
	@Ri	A←(A)+((Ri))	√	√	√	√	1	1
	#data	A←(A)+data	√	√	√	√	2	1
ADDC A,	Rn	A←(A)+(Rn)+(C)	√	√	√	√	1	1
	direct	A←(A)+(direct) +(C)	√	√	√	√	2	1
	@Ri	A←(A)+((Ri)) +(C)	√	√	√	√	1	1
	#data	A←(A)+data+(C)	√	√	√	√	2	1
INC	A	A←(A)+1	√	×	×	×	1	1
	Rn	Rn←(Rn)+1	×	×	×	×	1	1
	direct	direct←(direct)+1	×	×	×	×	2	1
	@Ri	(Ri)←((Ri))+1	×	×	×	×	1	1
	DPTR	DPTR←((DPTR))+1	×	×	×	×	1	2
DA A		BCD code adjustment	√	×	√	√	1	1
SUBB A	Rn	A←(A)-(Rn)-(C)	√	×	×	×	1	1
	direct	A←(A)-(direct)-(C)	√	√	√	√	2	1
	@Ri	(A)←(A)-((Ri))-(C)	√	√	√	√	1	1
	#data	A←(A)-data-(C)	√	√	√	√	2	1
DEC	A	A←(A)-1	√	×	×	×	1	1
	Rn	Rn←(Rn)-1	×	×	×	×	1	1
	direct	direct←(direct)-1	×	×	×	×	2	1
	@Ri	(Ri)←((Ri))-1	×	×	×	×	1	1
MUL AB		BA←(A)*(B),after performing multiplication, the low byte of the result is stored in A and the high byte is stored in B	√	√	×	0	1	4
DIV AB		A←(A)/(B) B←remainder	√	√	×	0	1	4
Note: When DA instruction is used, the adjustment rule is as follows: if the low 4 bits of Accumulator A is greater than 9 or AC=1, then A←A+06H; if the high 4 bits of accumulator A is greater than 9 or CY=1, then A←A+60H								

Logical operation instruction								
mnemonic sign		Function	Effect on flag bits				Number of bytes	Number of cycles
			P	OV	AC	CY		
CLR A		A←00H	√	×	×	×	1	1
CPL A		A←$\overline{(A)}$	√	×	×	×	1	1
ANL A,	Rn	A←(A)∧ (Rn)	√	×	×	×	1	1
	direct	A←(A)∧ (direct)	√	×	×	×	2	1
	@Ri	A←(A)∧ ((Ri))	√	×	×	×	1	1
	#data	A←(A)∧ data	√	×	×	×	2	1
ANL direct,	A	direct←(A)∧ (direct)	×	×	×	×	2	1
	#data	direct←(direct)∧ data	×	×	×	×	3	2
ORL A,	Rn	A←(A)∨ (Rn)	√	×	×	×	1	1
	direct	A←(A)∨ (direct)	√	×	×	×	2	1
	@Ri	A←(A)∨ ((Ri))	√	×	×	×	1	1
	#data	A←(A)∨ data	√	×	×	×	2	1
ORL direct,	A	direct←(direct)∨ (A)	×	×	×	×	2	1
	#data	direct←(direct)∨ data	×	×	×	×	3	2
XRL A,	Rn	A←(A)⊕ (Rn)	√	×	×	×	1	1
	direct	A←(A)⊕ (direct)	√	×	×	×	2	1
	@Ri	A←(A)⊕ ((Ri))	√	×	×	×	1	1
	#data	A←(A)⊕ data	√	×	×	×	2	1
XRL direct,	A	direct←(direct)⊕ (A)	×	×	×	×	2	1
	#data	direct←(direct)⊕ data	×	×	×	×	3	2

Loop and shift instructions							
mnemonic sign	Function	Effect on flag bits				Number of bytes	Number of cycles
		P	OV	AC	CY		
RL A	The content in A is shifted each bit to the left	×	×	×	×	1	1
RLC A	The contents of A are shifted each bit to the left with a Carry flag	√	×	×	√	1	1
RR A	The content in A is shifted each bit to the right	×	×	×	×	1	1
RRC A	The contents of A are shifted each bit to the right with a carry flag	√	×	×	√	1	1

Call, Return instructions							
mnemonic sign	Function	Effect on flag bits				Number of bytes	Number of cycles
		P	OV	AC	CY		

LCALL addr16		(PC)←(PC)+3,(SP)←(PC), (PC)←addr16	×	×	×	×	3	2
ACALL addr11		(PC)←(PC)+2,(SP)←(PC), (PC10~0)←addr11	×	×	×	×	2	2
RET		(PC)←((SP))	×	×	×	×	1	2
RETI		(PC)←((SP)) return from interrupt	×	×	×	×	1	2
Transfer instructions								
mnemonic sign		Function	Effect on flag bits				Number of bytes	Number of cycles
			P	OV	AC	CY		
LJMP addr16		PC←addr15~0	×	×	×	×	3	2
AJMP addr11		PC10~0←addr10~0	×	×	×	×	2	2
SJMP rel		PC←(PC)+rel	×	×	×	×	2	2
JMP @A+DPTR		PC←(A)+(DPTR)	×	×	×	×	1	2
JZ rel		PC←(PC)+2, If (A)=0,PC←(PC)+rel	×	×	×	×	2	2
JNZ rel		PC←(PC)+2, If (A)≠0,PC←(PC)+rel	×	×	×	×	2	2
JC rel		PC←(PC)+2, If(CY)=1,PC←(PC)+rel	×	×	×	×	2	2
JNC rel		PC←(PC)+2, If(CY)=0,PC←(PC)+rel	×	×	×	×	2	2
JB bit,rel		PC←(PC)+3, If (bit)=1,PC←(PC)+rel	×	×	×	×	3	2
JNB bit,rel		PC←(PC)+3, If (bit)=0,PC←(PC)+rel	×	×	×	×	3	2
JBC bit,rel		PC←(PC)+3, If (bit)=1, then bit←0, PC←(PC)+rel	×	×	×	×	3	2

CJNE	A, direct,rel	PC←(PC)+3,
		If (A) ≠direct, then PC(PC)+rel
		If(A)<(direct), then CY←1
			×	×	×	×	3	2

	A,#data,rel	PC←(PC)+3，
		If (A) ≠data, then PC(PC)+rel
		If (A)<(data), then CY←1
			×	×	×	×	3	2

	Rn,#data,rel	PC←(PC)+3， If (Rn) ≠data,	×	×	×	×	3	2

		PC←(PC)+rel If (Rn)<(data), then CY←1						
	@Ri,#data,rel	PC←(PC)+3, if((Ri))≠data, then PC←(PC)+rel. If ((Ri))<(data), then CY←1	×	×	×	×	3	2
DJNZ	Rn,rel	PC←(PC)+2,Rn←(Rn)-1, If (Rn) ≠0, then PC←(PC)+rel	×	×	×	×	2	2
	direct,rel	PC←(PC)+3, (direct)←(direct)-1. If (direct) ≠0, then PC←(PC)+rel	×	×	×	×	3	2

Stack, empty operation instructions

mnemonic sign	Function	Effect on flag bits				Number of bytes	Number of cycles
		P	OV	AC	CY		
PUSH direct	SP←(SP)+1,(SP)←(direct)	×	×	×	×	2	2
POP direct	direct←(SP),SP←(SP)-1	×	×	×	×	2	2
NOP	No Operation Performed	×	×	×	×	1	1

Bit manipulation instructions

mnemonic sign		Function	Effect on flag bits				Number of bytes	Number of cycles
			P	OV	AC	CY		
MOV	C,bit	CY←bit	×	×	×	√	2	1
	bit,C	bit←CY	×	×	×	×	2	1
CLR	C	CY←0	×	×	×	√	1	1
	bit	bit←0	×	×	×	×	2	1
SETB	C	CY←1	×	×	×	√	1	1
	bit	bit←1	×	×	×	×	2	1
CPL	C	$CY \leftarrow (\overline{CY})$	×	×	×	√	1	1
	bit	$bit \leftarrow (\overline{bit})$	×	×	×	×	1	1
ANL	C,bit	C←(C)∧(bit)	×	×	×	√	2	2
	C ,/bit	$C \leftarrow (C) \wedge (\overline{bit})$	×	×	×	√	2	2
ORL	C,bit	C←(C)∨(bit)	×	×	×	√	2	2

	C,/bit	C←(C)∨ ($\overline{bit}$)	×	×	×	√	2	2

Pseudo Instructions		
mnemonic sign	Format	Description
ORG	[Label:] ORG addr16	Specify the starting address of the label
EQU	Label EQU Value or label	Assign a value to the label
DB	[Label:] DB item or item list	Used to define a byte content of a cell or batch of cells in memory
DW	[Label:] DW item or item list	Used to define the contents of a 16-bit word consisting of two or more cells of memory
DS	[Label:] DS Expression	Specify to leave a number of storage cells starting from the label
BIT	label BIT address	Assigning bit addresses to label
END	END is placed at the end of the assembly language program to tell the assembler that the source program has come to an end.The source program that does not end with END will enter a dead loop	

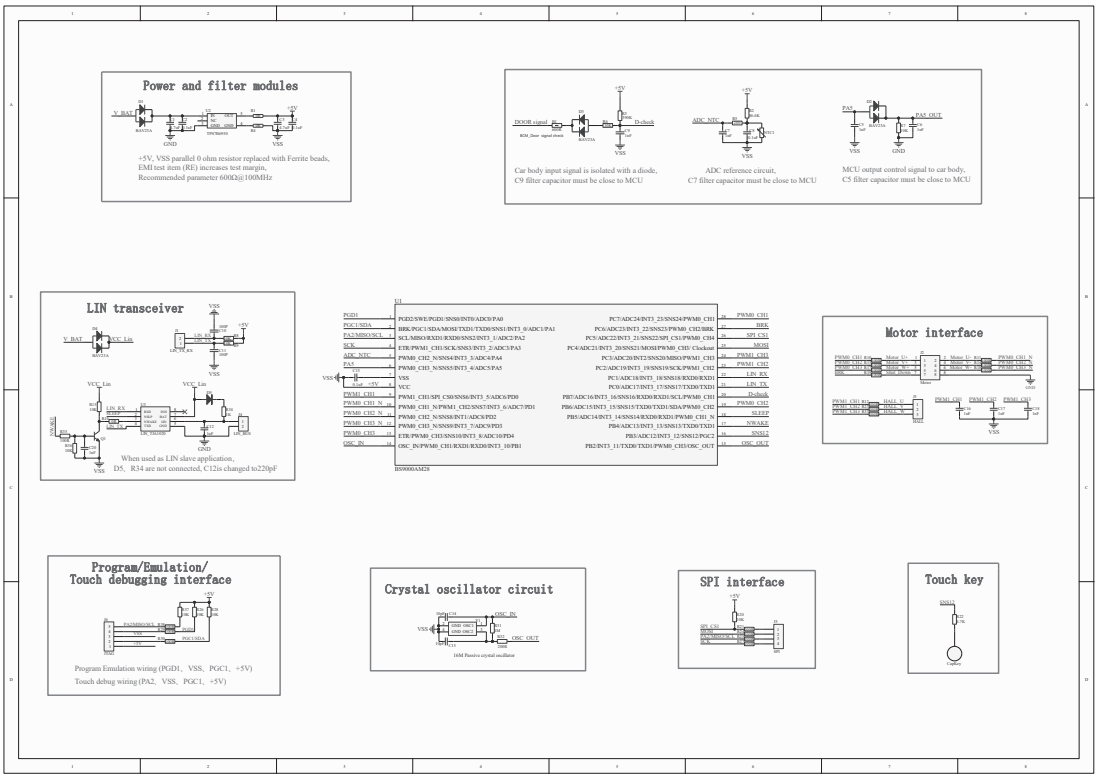
Table 0 .1 CPU Instruction Set

CPU-related Registers

SFR Register				
Address	Register Name	R/W	Reset Value	Description
0x81	SP	RW	0x07	Stack pointer
0x82	DPL	RW	0x00	Data Pointer *Low 8 bit*
0x83	DPH	RW	0x00	Data Pointer High *8 bit*
0x87	PCON	RW	0x00	Low power mode select register
0xE0	ACC	RW	0x00	Accumulator
0xF0	B	RW	0x00	B Register

Table 16.3 CPU SFR Register

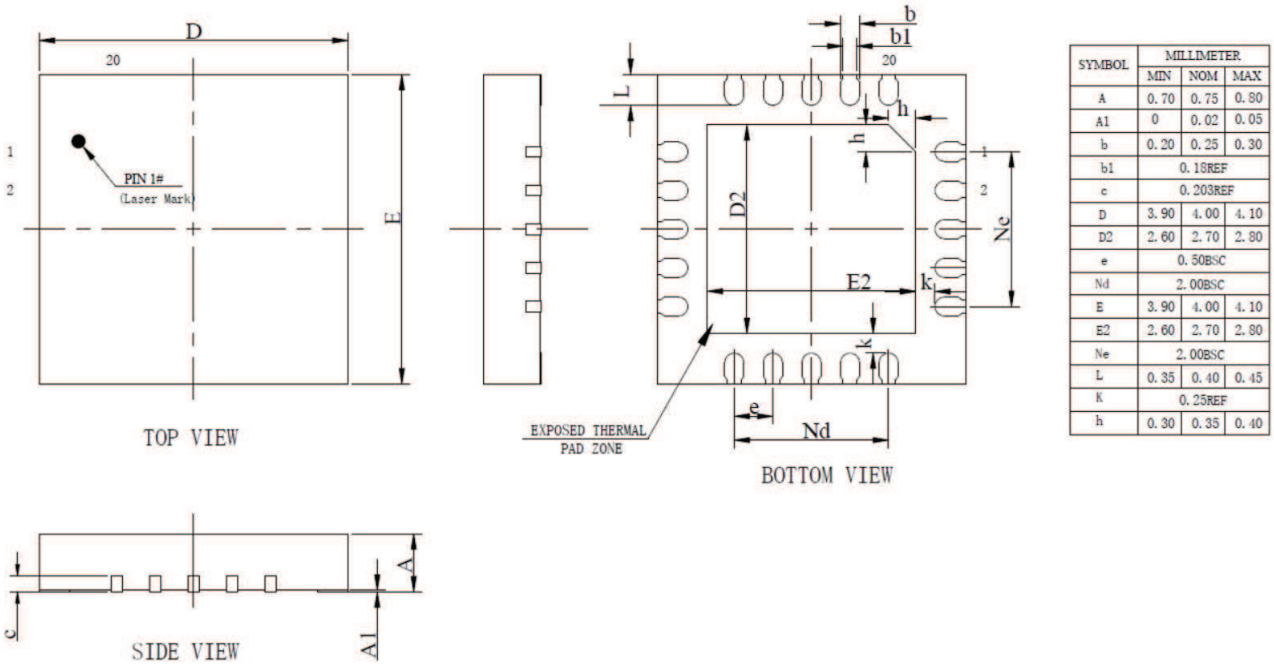
17 Application Circuit For Reference



Note: The above circuit is designed for reference only.

18 Package Information

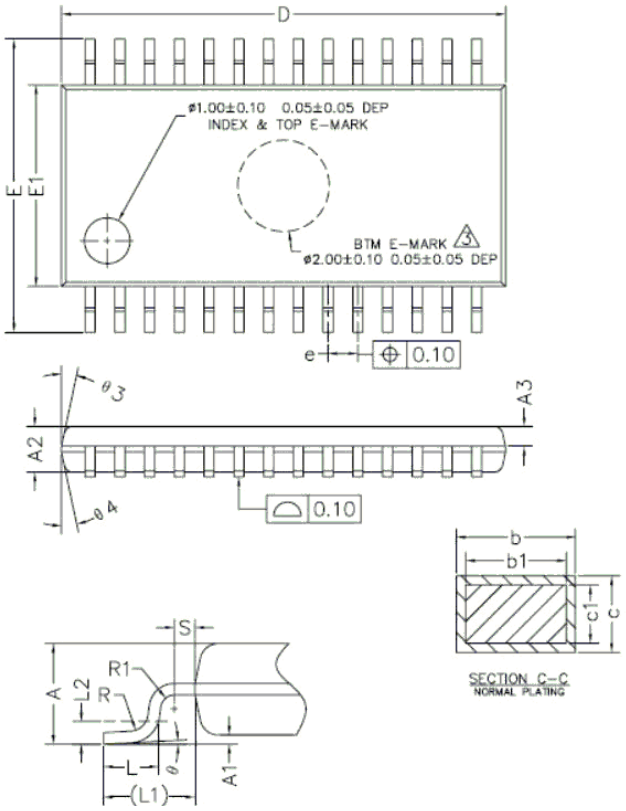
18.1 QFN20

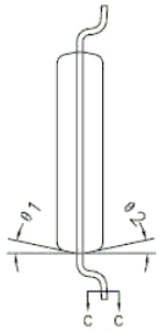


SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.20	0.25	0.30
b1		0.18REF	
c		0.203REF	
D	3.90	4.00	4.10
D2	2.60	2.70	2.80
e		0.50BSC	
Nd		2.00BSC	
E	3.90	4.00	4.10
E2	2.60	2.70	2.80
Ne		2.00BSC	
L	0.35	0.40	0.45
K		0.25REF	
h	0.30	0.35	0.40

Figure 18.1 QFN20 Packaging Information Diagram

18.2 TSSOP28





COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

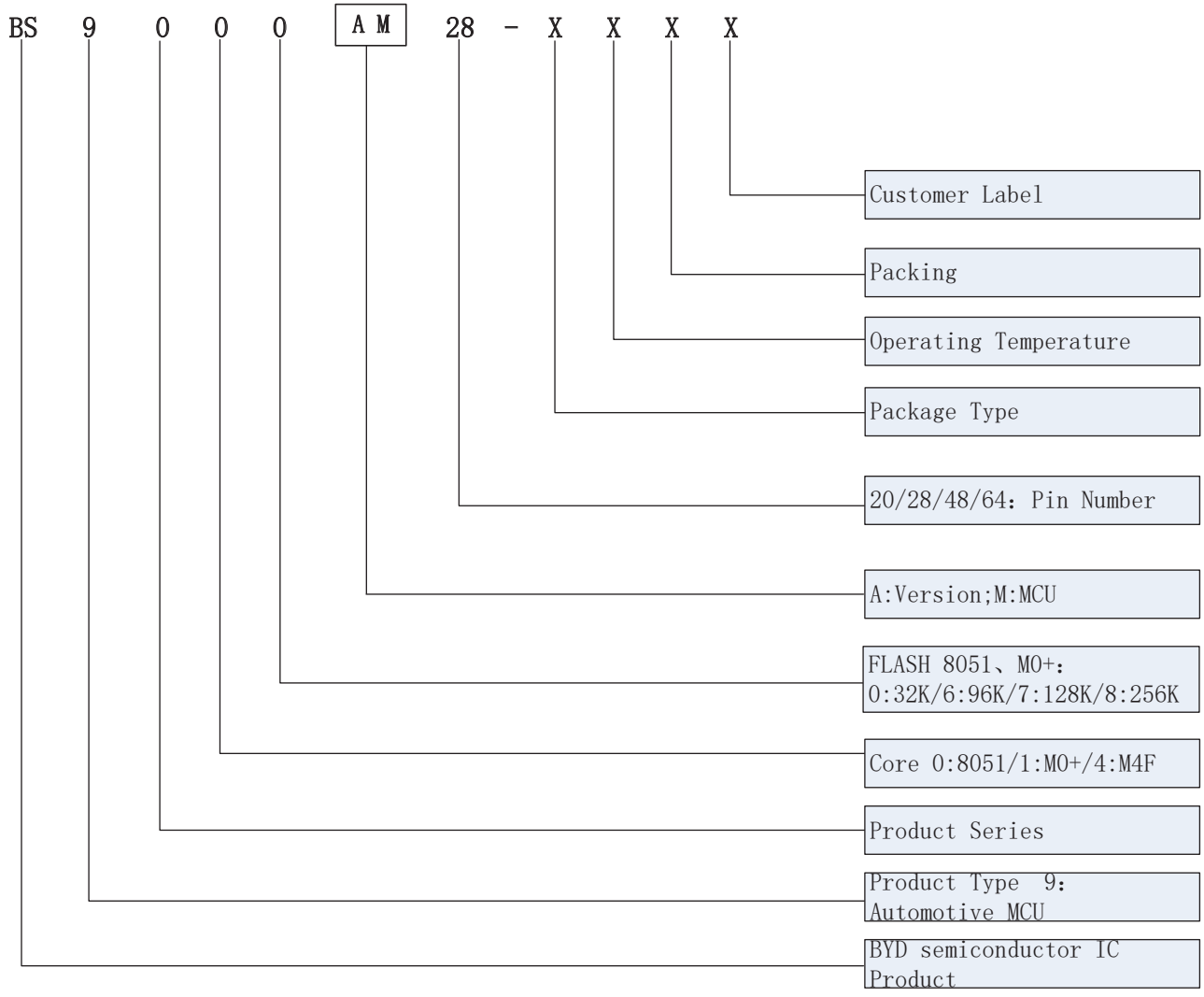
SYMBOL	MIN	NOM	MAX
A	–	–	1.20
A1	0.05	–	0.15
A2	0.90	1.00	1.05
A3	0.34	0.44	0.54
b	0.20	–	0.29
b1	0.19	0.22	0.25
c	0.13	–	0.18
c1	0.12	0.13	0.14
D	9.60	9.70	9.80
E	6.20	6.40	6.60
E1	4.30	4.40	4.50
e	0.55	0.65	0.75
L	0.45	0.60	0.75
L1		1.00REF	
L2		0.25BSC	
R	0.09	–	–
R1	0.09	–	–
S	0.20	–	–
θ	0°	–	8°
θ1	10°	12°	14°
θ2	10°	12°	14°
θ3	10°	12°	14°
θ4	10°	12°	14°

Figure 18.2 TSSOP28 Package

19 Ordering Information

Package Type	Temperature Range		Packing
S: SOP	Automotive-grade	A:-40℃ to +150℃	B: Tape & Reel
A: SSOP		B: -40℃ to +125℃	L: Tube
T: TSSOP		C: -40℃ to +105℃	T: Tray
M: MSSOP		D: -40℃ to +85℃	-
L: LQFP	Industrial-grade	K: -40℃ to +85℃	-
Q: QFN		J: -40℃ to +105℃	-
B: BGA		L: -40℃ to +125℃	-
D: DIP	Consumer-grade	P: -25℃ to +70℃	-
-		Q: 0℃ to +70℃	-

See a schematic diagram below.



20 Revision History

Date	Description	Complied by	REV
2022-05-30	Initial release	Liu Hong	V1.0

21 Disclaimer

BYD Semiconductor makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. BYD Semiconductor does not make any commitment to update the information contained herein. Please contact BYD FAE or authorized agents to obtain new version of this document.

BYD Semiconductor will do its best to guarantee the high quality and stability of products. Nevertheless, damages may be caused to components or devices due to its intrinsic electrical and physical susceptibility, which may result in breakdown or failure. It is the user's liability to design a safe and stable system environment following the safety requirements when using BYD products. Users can take measures to avoid the possible accidents, fire and public injuries by the removal of the extra devices, failure and fire prevention. When using the product, users need to follow the operation steps specified in the latest specification.

The product of BYD Semiconductor contained herein is designed for electrically generic application, such as computer, PC, office equipments and measurement instruments, industrial machines and home applicances. However, the products are not suitable for use in a particular devices that require ultra-high stability and quality, which may result in personal injury or death. BYD semiconductor products are not designed for use in nuclear facilities, airplanes or aerospace devices, transportation, traffic signal, combustion control equipment and medical or life-saving devices or system. BYD Semiconductor disclaims any and all liability including loss and damage arising from use in above-mentioned applications.