LIN Interface General Purpose MCU

Features

- Operating voltage range: 6V~18V, load dump voltage: 45V
- 32-bit MCU
- Maximum operating frequency: 16MHz
- 64KB FLASH, with erase protection function
- 16KB SRAM memory
- 4-channel 16bit PWM current sinks, maximum 60mA/channel
- Support LS junction voltage (0V~8V) detection, support LS temperature compensation
- 12 general GPIO ports
  - Support PWM mode
  - Support external LIN transceiver multiplexing function
- Integrated LIN transceiver, support automatic addressing function
- Support SAE J2602/LIN2.x protocol LIN slave interface
- 1 WDT and 1 WWDG
- 2 Basic Timers, 1 General-Purpose Timer, 1 Wakeup Timer
- 1 I^2C, 1 SPI, 1 UART
- 12-bit ADC converter, supporting 13 channels
  - Optional reference voltage (1.2V/2.4V/3.3V)
  - Temperature sensor
  - VBAT voltage detection
  - LS junction voltage detection
- Support for SWD debugging
- Support Power on Reset (POR) and Brown out Reset (BOR) function
- Operation range: -40~125°C
- Package: QFN 5x5-32L

General Description

AW32F020QNR-Q1 is a chip designed specifically for automotive window motor controllers, which can improve the convenience of debugging automotive window controls.

The AW32F020QNR-Q1 integrates a 32-bit MCU. The MCU provides superior computational performance and an efficient interrupt system, while the peripheral devices use a smaller number of pins and lower power consumption.

AW32F020QNR-Q1 integrates 64KB FLASH memory with erase/write protection function. The LIN communication module is integrated to support the transceiver function of LIN communication.

Integrated 12-bit ADC for LS junction voltage and temperature detection, supporting 13 channels, VREF voltage can be configured to 1.2V/2.4V/3.3V.

Support 4 high voltage current sinks, the maximum current of each channel is up to 60mA. Support 16-bit PWM dimming, support 80Hz~250Hz dimming frequency.

Two basic timers, one general-purpose timer and one WDT are integrated. 12 general GPIO ports are integrated to support PWM mode and external LIN receiver interconnection. Support one hardware I2C or SPI or UART.

The SWD debug interface is integrated with built-in Power on Reset (POR) and Brown out Reset (BOR) functions to effectively protect the chip.

AW32F020QNR-Q1 is packaged in QFN 5x5-32L.

Applications

Automotive motor control

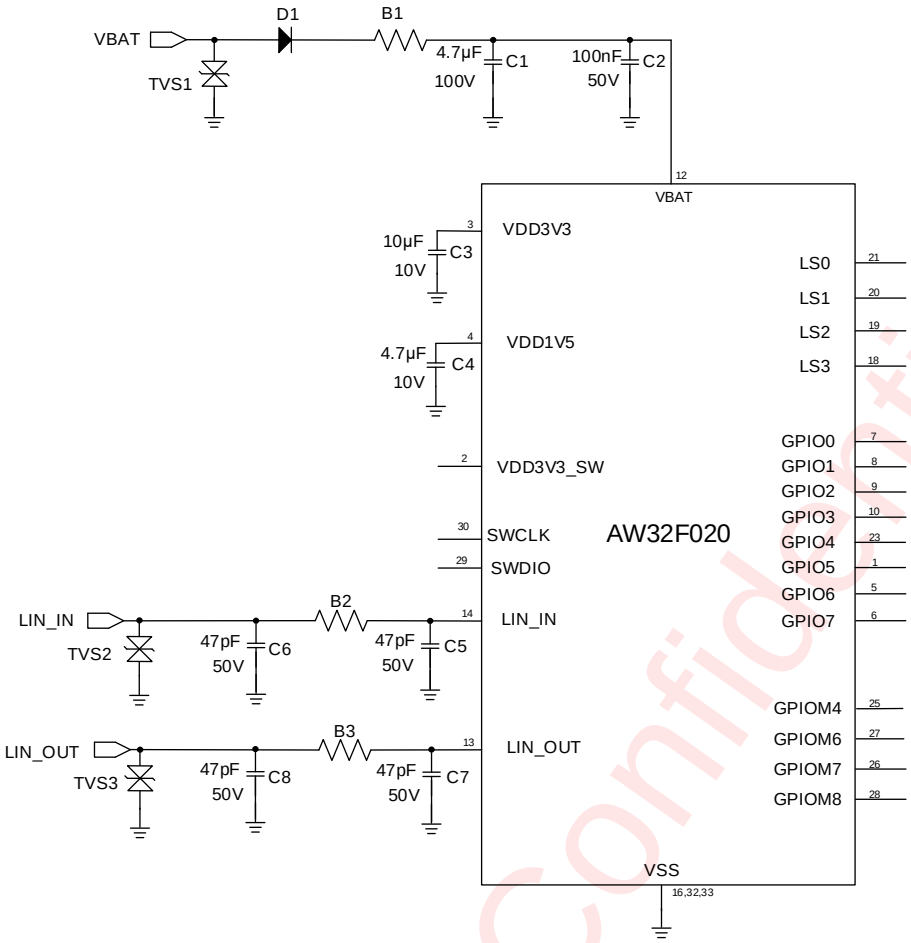
Electric window control

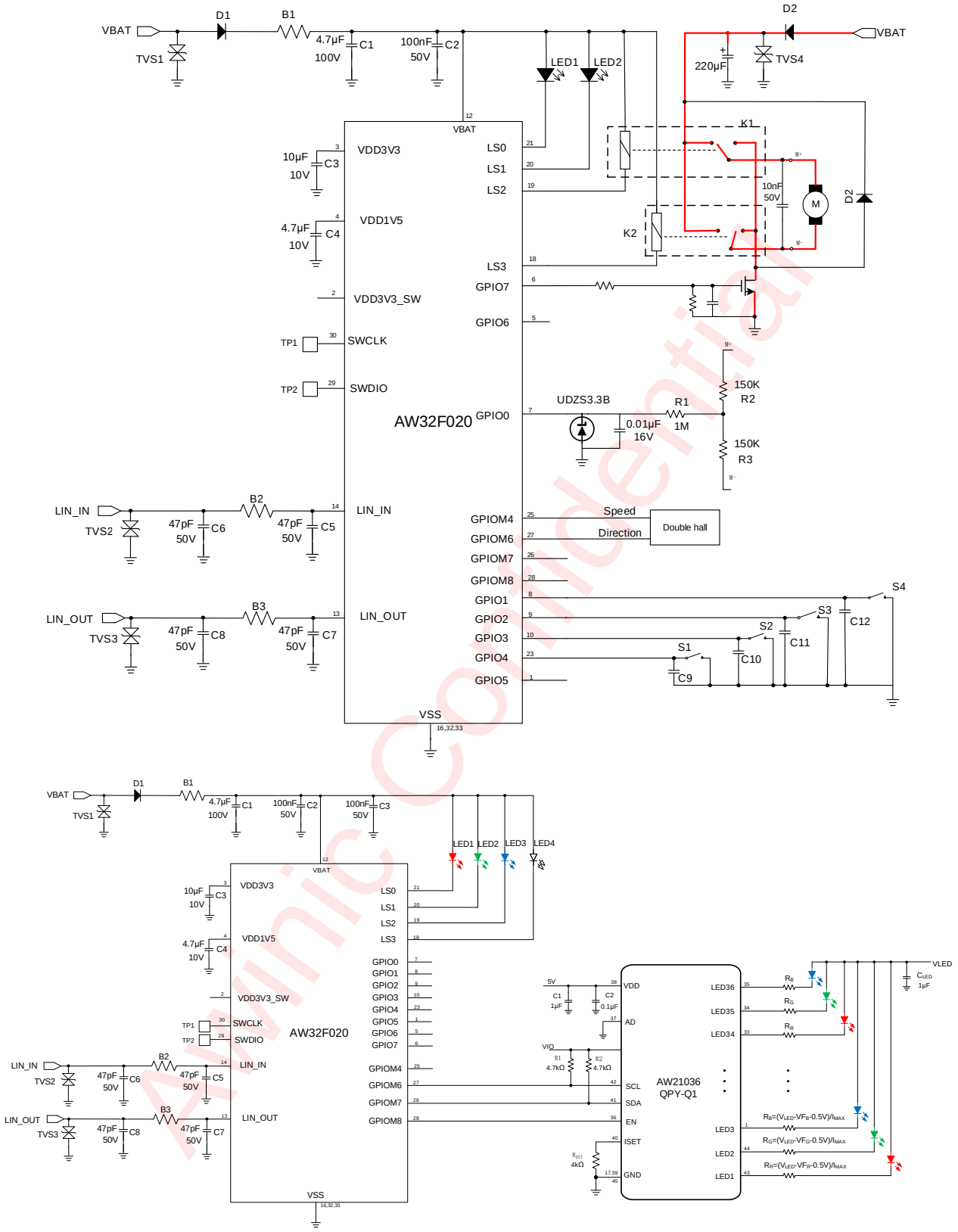
Electric seat control

Automotive LED driver

Door locks

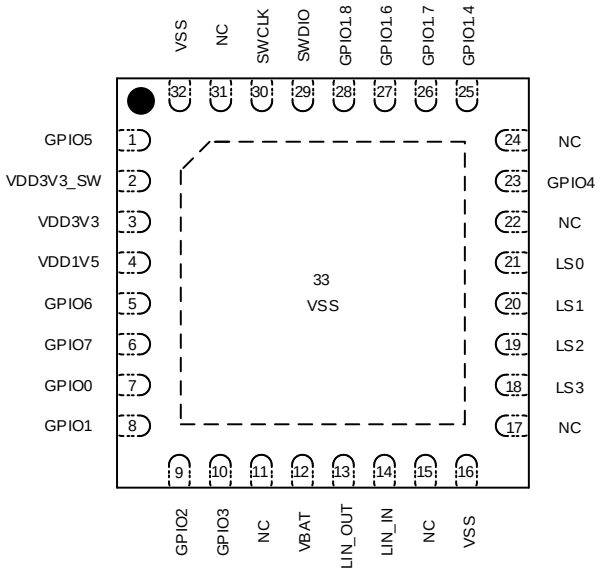
Typical Application Circuit



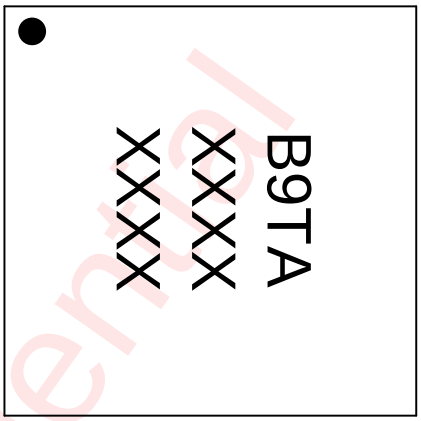


Pin Configuration And Top Mark

AW32F020QNR-Q1
(Top View)



AW32F020QNR-Q1 Marking
(Top View)



B9TA - AW32F020QNR-Q1
XXXX/XXXX - Production Tracing Code

Figure 1 Pin Configuration and Top Marking

Pin Definition

No.	NAME	DESCRIPTION
1	GPIO5	General purpose IO. ADC_IN/PWM_OUT
2	VDD3V3_SW	LDO output 3.3V, 10mA max. Closed in sleep mode.
3	VDD3V3	Connect to the external 10μF capacitor.
4	VDD1V5	Connect to the external 4.7μF capacitor.
5	GPIO6	General purpose IO.
6	GPIO7	General purpose IO.
7	GPIO0	General purpose IO. ADC_IN/PWM_OUT.
8	GPIO1	General purpose IO. ADC_IN/PWM_OUT.
9	GPIO2	General purpose IO. ADC_IN/PWM_OUT.
10	GPIO3	General purpose IO. ADC_IN/PWM_OUT.
11	NC	No connect.
12	VBAT	Supply voltage.
13	LIN_OUT	J2602 LIN 2.x.
14	LIN_IN	J2602 LIN 2.x.
15	NC	No connect.
16	VSS	Ground.

17	NC	No connect.
18	LS3	Low-side switch. PWM_OUT.
19	LS2	Low-side switch. PWM_OUT.
20	LS1	Low-side switch. PWM_OUT.
21	LS0	Low-side switch. PWM_OUT.
22	NC	No connect.
23	GPIO4	General purpose IO. ADC_IN/PWM_OUT.
24	NC	No connect.
25	GPIOM4	General purpose IO, TIM3_CH1/SPI_MISO/PWM_OUT.
26	GPIOM7	General purpose IO, TIM3_CH3/UART_TX/I²C_SDA/SPI_MOSI/PWM_OUT.
27	GPIOM6	General purpose IO, TIM3_CH2/UART_RX/I²C_SCL/SPI_SCK/PWM_OUT.
28	GPIOM8	General purpose IO, TIM3_CH4/SPI_CS/PWM_OUT.
29	SWDIO	Debugger data. Integrated weak pull up.
30	SWCLK	Debugger clk. Integrated weak pull up.
31	NC	No connect.
32	VSS	Ground.
33	VSS	Ground.

FUNCTIONAL BLOCK DIAGRAM

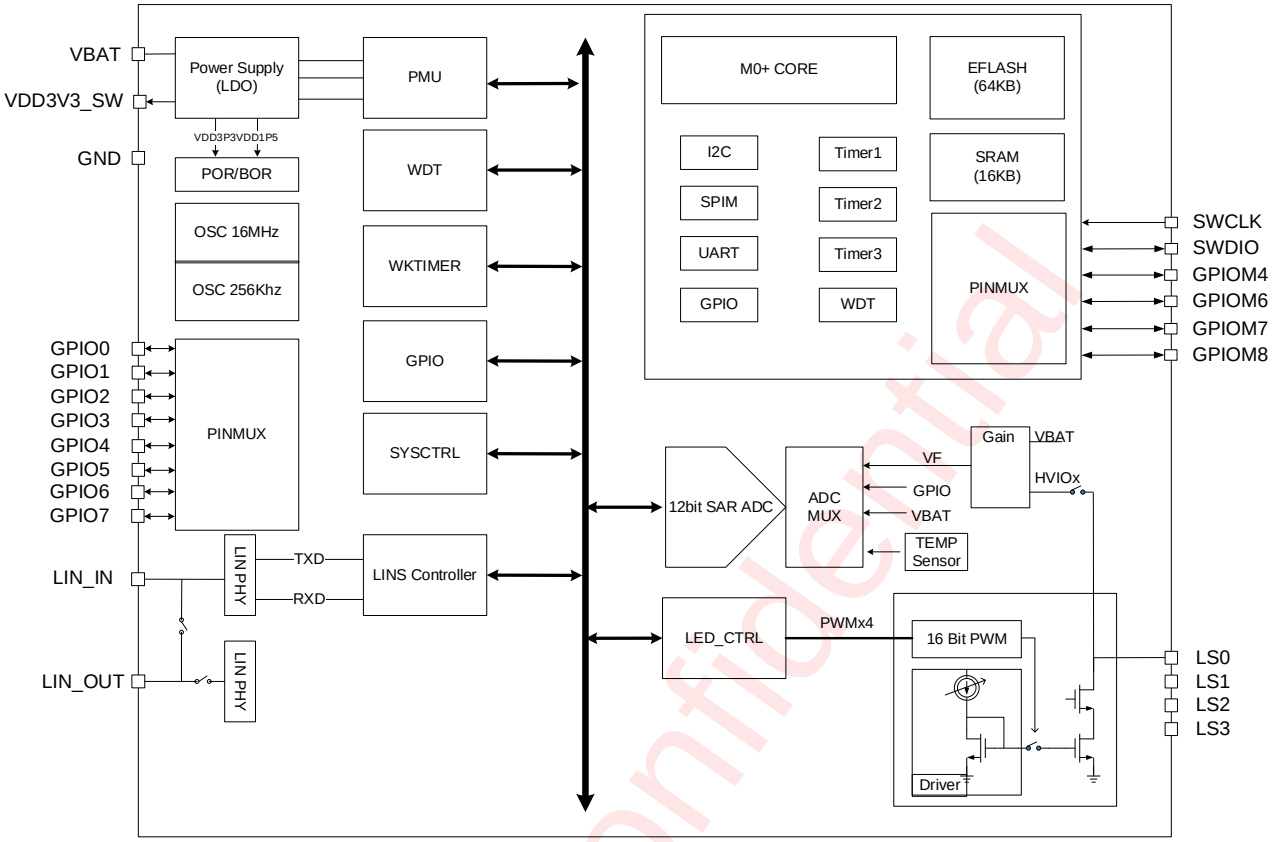


Figure 2 Function Block Diagram

TYPICAL APPLICATION CIRCUITS

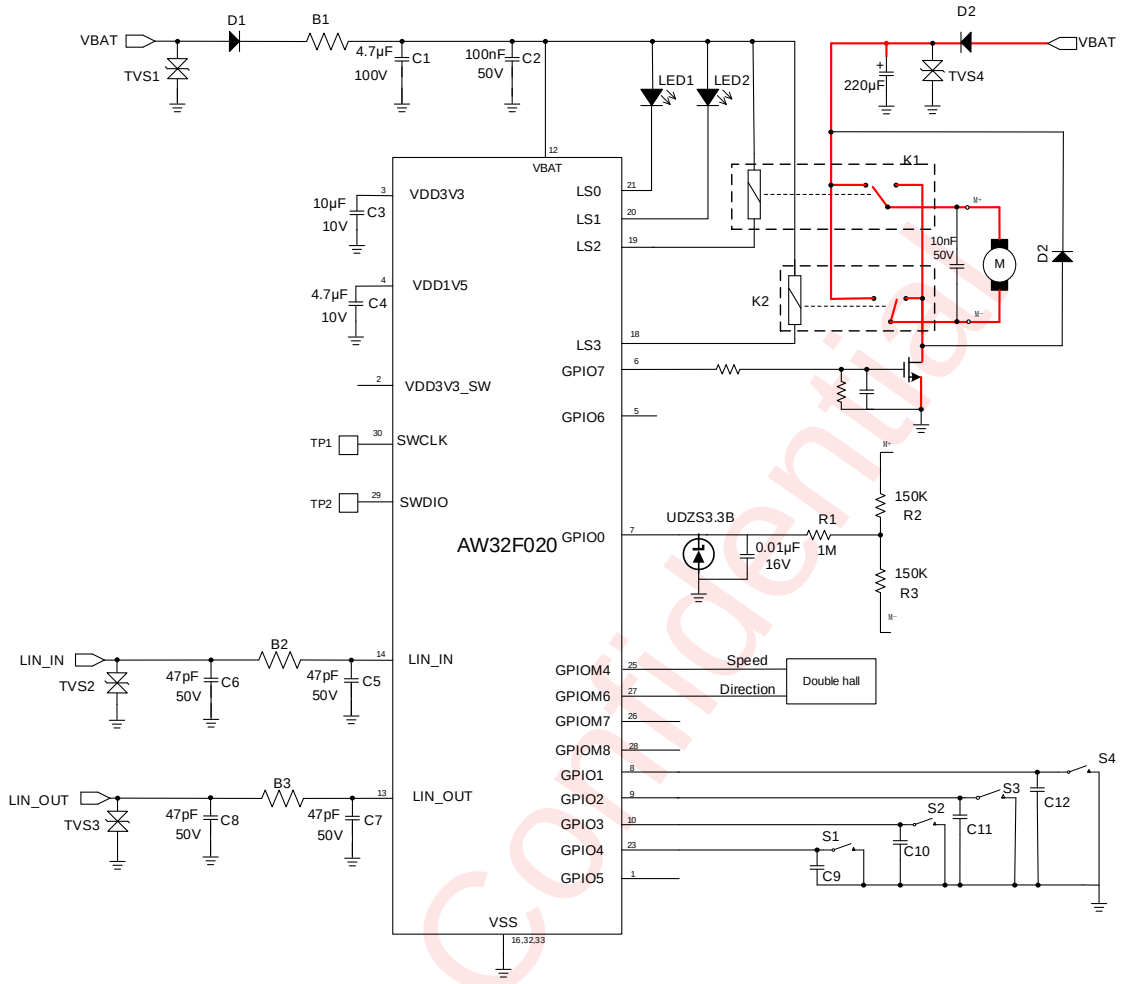


Figure 3 Typical Application Circuit

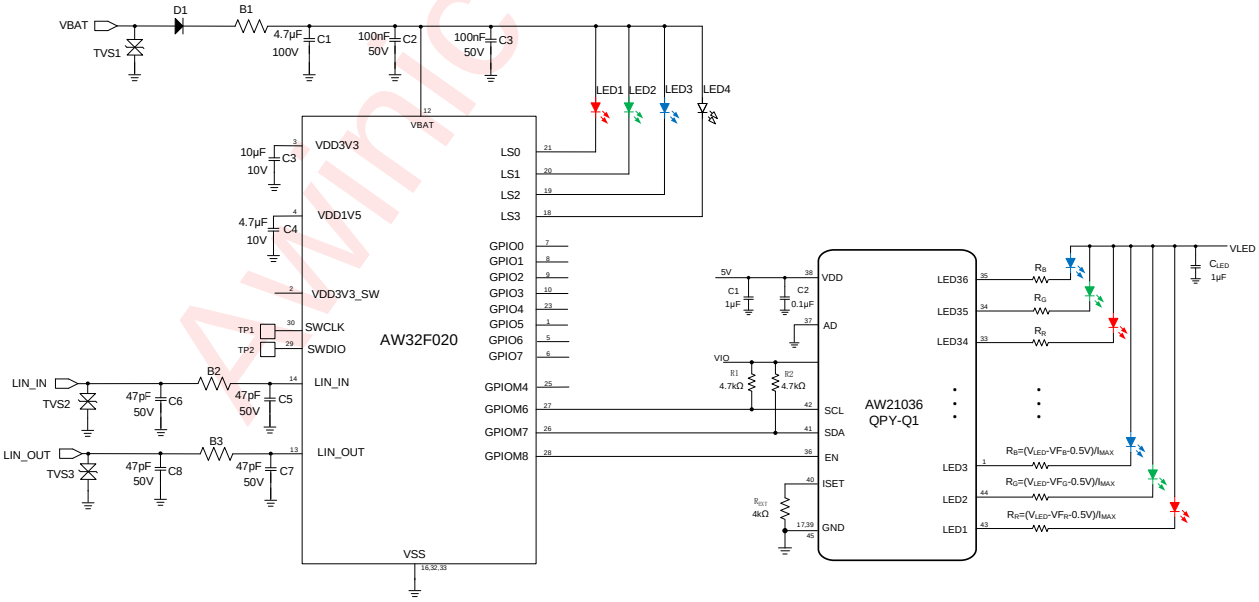


Figure 4 Typical Application Circuit

Ordering Information

Part Number	Temperature	Package	Marking	Moisture Sensitivity Level	Environmental Information	Delivery Form
AW32F020QNR-Q1	-40℃～125℃	QFN 5mm X 5mm-32L	B9TA	MSL3	ROHS+HF	6000 units/ Tape and Reel

ABSOLUTE MAXIMUM RATINGS(NOTE1)

PARAMETERS		RANGE
VBAT	500ms	-0.3V to 45V
	5min	-0.3V to 28V
	5ms	-1.1V
	20ns	-4V
	ISO 7637-2 pulse 1, VBAT=13.5V, TA=(23±5)℃	-112V
	ISO 7637-2 pulse 2a, VBAT=13.5V, TA=(23±5)℃	55V
	ISO 7637-2 pulse 2b, VBAT=13.5V, TA=(23±5)℃	10V
	ISO 7637-2 pulse 3a, VBAT=13.5V, TA=(23±5)℃	-165V
	ISO 7637-2 pulse 3b, VBAT=13.5V, TA=(23±5)℃	112V
LIN_IN, LIN_OUT	500ms	-40V to 40V
	ISO 7637-3 pulse +2a, VBAT=13.5V, TA=(23±5)℃	5V
	ISO 7637-3 pulse -2a, VBAT=13.5V, TA=(23±5)℃	-5v
	ISO 7637-3 pulse 3a, VBAT=13.5V, TA=(23±5)℃	-80V
	ISO 7637-3 pulse 3b, VBAT=13.5V, TA=(23±5)℃	60V
LSx	500ms	-0.3V to 45V
	5min	-0.3V to 28V
	5ms	-1.1V
	20ns	-4V
VBAT, LIN_IN, LIN_OUT, LSx		-0.3V to 24V
GPIOx, SWDCLK, SWDIO		-0.3V to 3.6V
Junction-to-ambient thermal resistance θ_{JA}		52.3°C /W
Maximum operating junction temperature T_{JMAX}		150°C
Storage temperature T_{STG}		-55°C to 150°C
Lead temperature (soldering 10 seconds)		260°C
ESD(Including CDM HBM)(NOTE 2)		
VBAT/LIN_IN/LIN_OUT to GND HBM (Test condition: AEC-Q100-002-RevE)		±4kV
Other PINS HBM (Test condition: AEC-Q100-002-RevE)		±2kV
CDM (Test condition: AEC-Q100-011-RevD)		±750V
Latch-Up		

Test condition: AEC_Q100-004	+IT: 450mA -IT: -450mA

*NOTE1: Conditions out of those ranges listed in "absolute maximum ratings" may cause permanent damages to the device. In spite of the limits above, functional operation conditions of the d*evice should within the ranges listed in "recommended operating conditions". Exposure to absolute-maximum-rated conditions for prolonged periods may affect device reliability.

NOTE2: The human body model is a 100pF capacitor discharged through a 1.5kΩ resistor into each pin.

Recommended Operating Conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{BAT}	Input voltage	6	13.5	18	V
C_1	Input capacitance connected to VBAT	2.2	4.7	10	μF
C_2	Input capacitance connected to VBAT		0.1		μF
C_{3V3}	Capacitance connected to VDD3V3	4.7		10	μF
C_{1V5}	Capacitance connected to VDD1V5	1		4.7	μF
T_A	Operating free-air temperature range	-40°	25	125	°C

ELECTRICAL CHARACTERISTICS

VBAT=6V to 18V, T_A=25°C for typical values (unless otherwise noted)

PARAMETER	TEST CONDITION	MIN	TYP	MAX	UNIT
Operation Conditions					
VBAT		6	13.5	18	V
IO Supply(VDD3V3)		2.97	3.3	3.63	V
ASIC Core Supply		2.97	3.3	3.63	V
MCU Core Supply	MCU Core Supply including SRAM and Flash	1.4	1.55	1.65	V
FLASH Memory					
Sector Endurance		20k			cycles
Data Retention	@25°C	100			Years
Data Retention	@85°C	25			Years
SRAM					
Min Retention Voltage	Minimum Retention Voltage below which SRAM data are not guaranteed.	1.35			V
CLOCKs					
System RC Oscillator Frequency			16		MHz
System RC Oscillator Accuracy	16MHz	-3		3	%
System RC Oscillator start up time			10		μs
Auxiliary system clock	Used in sleep mode		256		kHz
Auxiliary system clock Accuracy		-10		10	%
POR/BOR					
POR power-on detection threshold	Triggered by VDD3V3 power-on	2.1	2.4	2.7	V
POR power-on detection hysteresis			0.2		V
Under Voltage Threshold of BOR	Triggered by VDD3V3 power-off, user configurable	2.4	2.55	2.7	V
		2.5	2.65	2.8	
		2.6	2.75	2.9	
		2.7	2.85	3	
		2.8	2.95	3.1	
Battery Monitor					
Under Voltage	Triggered by VBAT power-off,	4.5	5.0	5.5	V

PARAMETER	TEST CONDITION	MIN	TYP	MAX	UNIT
Threshold	generates interrupt to MCU(except in sleep mode), user configurable				
		5.5	6.0	6.5	V
		6.5	7.0	7.5	V
		7.5	8.0	8.5	V
		8.5	9.0	9.5	V
Under Voltage hysteresis	The under voltage threshold is set to 5V, user configurable	0.01	0.125	0.255	V
Under Voltage Digital debounce time	User configurable, the step size is 62.5ns or 62.5μs			16.32	ms
Over Voltage Threshold	Triggered by VBAT power-on, Generates interrupt to MCU(except in sleep mode), user configurable	21	22.5	24	V
		23.5	25	26.5	
		26	27.5	29	
		28.5	30	31.5	
		31	32.5	34	
		33.5	35	36.5	
		36	37.5	39	
Over Voltage Hysteresis	The over voltage threshold is set to 25V, user configurable	0.04	0.714	1.624	V
Over Voltage Digital debounce time	User configurable, the step size is 62.5ns or 62.5μs			16.32	ms
Current Source (LS)					
LS Dropout Voltage	The voltage when the current drops to 90%(60mA)			1.6	V
Sink Current	VBAT>6V	0.12		60	mA
Sink Current step size			120		μA
Sink Current Error	The temperature is 25°C	-7		+7	%
Temperature Drift			-0.025		%/°C
Over Temperature Monitor					
Overtemp Threshold	Generates interrupt or reset to MCU	90		165	°C
Overtemp hysteresis		-10	-14		°C
Temperature Sensor					
Temperature detection range	The MCU reads ADC data related to temperature detection	-40		150	°C
Temperature detection Accuracy		-10		+10	°C
Active current			8		μA
Differential Amplifier (LS VFW measurement)					

PARAMETER	TEST CONDITION	MIN	TYP	MAX	UNIT
Input Voltage(Junction voltage range of LS lamp)	<4V @Gain=1/4	0		4	V
	>4V @Gain=1/8	4		8	V
Output Voltage Range		0		1	V
Output Voltage Relative Error				1	%
Gain	User configurable		1/4 or 1/8		
Wake Up					
T_{WAKEUP}	LIN_IN/LIN_OUT, User configurable	30	150	200	μs
Wake Up Timer	Wakeup Time =TimerLoadCount/16kHz, TimerLoadCount = 0x0~0x1FFFFFF, the default value is 0x1FFFFFF	0		2097152	ms
ASIC Watchdog timer					
Timeout	User configurable	0.128		16	s
SAR ADC					
Resolution			12		Bits
Conversion Speed	17 cycles per conversion (4 cycles for sampling and 13 cycles for conversion)			200	ks/s
ADC Clock	16MHz RC clock divided by 4			4	MHz
INL	Guaranteed by design	-2		2	LSB
DNL	Guaranteed by design	-1		1	LSB
Reference voltage[1]	Post Calibration	1.19	1.20	1.21	V
LIN(VBAT=8V~16V, VBUS is the voltage of LIN bus, VSUP is the chip's power supply voltage)					
Supply Voltage	supply voltage range	6	13.5	18	V
I_{BUS_LIM}	Current Limitation for Driver dominant state driver on VBUS = VBAT=16V	40		200	mA
R_{slave}	Lin Slave Pull-up	20	30	60	kΩ
BUS_PAS_dom	Driver off, VBUS=0, VBAT=12V	-1			mA
$I_{BUS\_PAS\_rec}$	Driver off, VBUS>VBAT 8V<VBAT<16V, 8V<VBUS<16V			20	μA
$I_{BUS\_no\_GND}$	Control unit disconnected from Ground, GND = VSUP 0V<VBUS<16V, VBAT = 12V, Loss of local ground must not affect communication in the residual network. LIN 2.2A	-1		+1	mA
$I_{BUS\_no\_BAT}$	VBAT disconnected 0<VBUS<16V,			100	μA

PARAMETER	TEST CONDITION	MIN	TYP	MAX	UNIT
	VBAT=0V LIN 2.2A Node has to sustain the current that can flow under this condition. Bus must remain operational under this condition.				
Device Bus Leakage current VBAT disconnected	0V<VBUS<18V, VBAT=VGND=0V J2602	-23		23	μA
Device Bus Leakage Current Ground Disconnected	VBAT= VGND=12V, 0V<VBUS<18V J2602	-100		100	μA
BUS_VOL Transmitter dominant voltage	Load 500Ohms, driver open drain active			0.2* VBUS	V
BUS_VOH Transmitter recessive voltage	Driver open drain high impedance	0.8* VBUS			V
C_{LIN}	LIN pin input capacitance			35	pF
VBUSrec	Receiver recessive state	0.6* VBUS			V
VBUSdom	Receiver dominant state			0.4* VBUS	V
V_{hys}	Receiver hysteresis Vhys = VBUSrec -VBUSdom			0.175* VBUS	V
VBUS_CNT	Center point Receiver VBUS_CNT =(VBUSrec+VBUSdom)/2	0.475* VBUS	0.5* VBUS	0.525* VBUS	V
Trx_pdr	Rising edge propagation delay of receiver C_{RXD} load 20pF (RX output of transceiver, internal node, access in test mode)			6	μs
Trx_pdf	Falling edge propagation delay of receiver C_{RXD} load 20pF (RX output of transceiver, internal node, access in test mode)			6	μs
Trx_sym	symmetry of receiver propagation delay Trx_sym= Trx_pdf- Trx_pdr	-2		2	μs
LIN Timing parameters					
D1 Duty Cycle (20kbits/s)	THRec(max) = 0.744 x VSUP; THDom(max) = 0.581 x VSUP; VSUP = 7.0V...16V; tBit = 50μs; D1 = tBus_rec(min) / (2 x tBit)	0.396			-
D2 Duty Cycle (20kbits/s)	THRec(min) = 0.422 x VSUP; THDom(min) = 0.284 x VSUP; VSUP = 7.6V...16V; tBit = 50μs;			0.581	-

PARAMETER	TEST CONDITION	MIN	TYP	MAX	UNIT
	D2 = tBus_rec(max) / (2 x tBit)				
D3 Duty Cycle (10.4kbits/s)	THRec(max) = 0.778 x VSUP; THDom(max) = 0.616 x VSUP; VSUP = 7.0V...16V; tBit = 96μs; D3 = tBus_rec(min) / (2 x tBit)	0.417			-
D4 Duty Cycle (10.4kbits/s)	THRec(min) = 0.389 x VSUP; THDom(min) = 0.251 x VSUP; VSUP = 7.6V...16V; tBit = 96μs; D4 = tBus_rec(max) / (2 x tBit)			0.59	-
$t_{Bus\_rec(max)}$- $t_{Bus\_dom(min)}$	Δt3, 10.4kbs operation, low speed mode, J2602			15.9	μs
$t_{Bus\_dom(max)}$- $t_{Bus\_rec(min)}$	Δt4, 10.4kbs operation, low speed mode, J2602			17.28	μs
GPIOs					
GPIO V_{IL}	Input Low Voltage			0.3* VDD3V3	V
GPIO V_{IH}	Input High Voltage	0.7* VDD3V3			V
GPIO I_{OL}	Max load current with output voltage=VOL			10	mA
GPIO I_{OH}[2]	Max load current with output voltage=VOLH			10	mA
GPIO V_{OL}	Output Low Voltage			0.4	V
GPIO V_{OH}	Output High Voltage	2.4			V
GPIO P_U	Pull Up Resistance		50		kΩ
GPIO P_D	Pull Down Resistance		50		kΩ
GPIOMs					
GPIOM V_{IL}	Input Low Voltage			0.3* VDD3V3	V
GPIOM V_{IH}	Input High Voltage	0.7* VDD3V3			V
GPIOM I_{OL}	Max load current with output voltage=VOL			5	mA
GPIOM I_{OH}[2]	Max load current with output voltage=VOLH			5	mA
GPIOM V_{OL}	Output Low Voltage			0.4	V
GPIOM V_{OH}	Output High Voltage	2.4			V
GPIOM P_U	Pull Up Resistance		20		kΩ
GPIOM P_D	Pull Down Resistance		20		kΩ

PARAMETER	TEST CONDITION	MIN	TYP	MAX	UNIT
SWCLK, SWDIO					
SWD V_{IL}				0.3* VDD3V3	V
SWD V_{IH}		0.7* VDD3V3			V
SWD V_{OL}				0.4	V
SWD V_{OH}		2.4			V
SWD P_U			20		kΩ

Note1: Minimum and/or maximum limit is guaranteed by design and by statistical analysis of device characterization data. The specification is not guaranteed by production testing.

Note2: The sum of max load current of all GPIO/GPIOM and VCC3V3_SW is 35mA.

TYPICAL CHARACTERISTICS

CURRENT CONSUMPTION

Mode	Conditions	Min	Typ	Max	Unit
Normal	Ta=85℃, VBAT=18V, RCO=16MHz, full functionality: MCU running, no flash write, LS OFF, ADC ON, VBAT and TEMP monitor ON, WDT ON.			10	mA
Sleep	Main regulator (3.3V) ON, Load dump protection active. Ta=85degC max, VBAT=13.5V Overvoltage/Undervoltage detection, PWM, LS driver, Temp_sensor and ADC are OFF except one LIN RX on and GPIO toggling and wake up timer.	15	35	60	μA

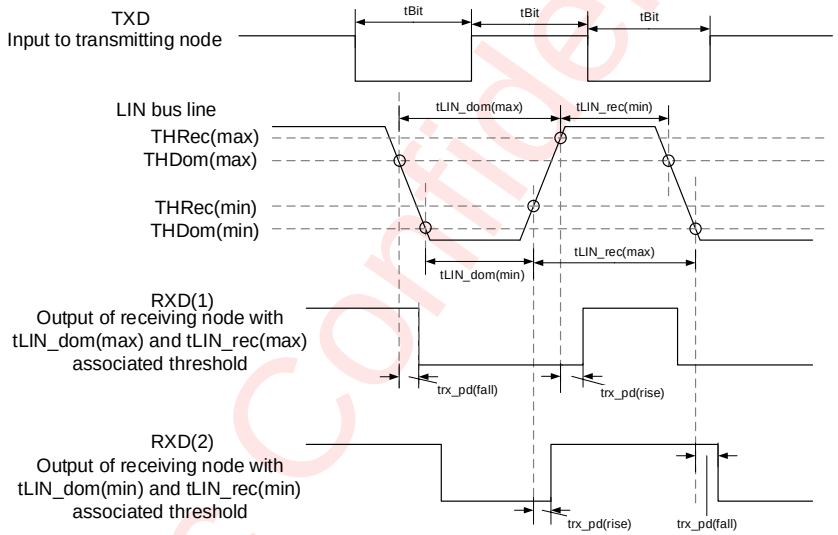


Figure 5 LIN timing

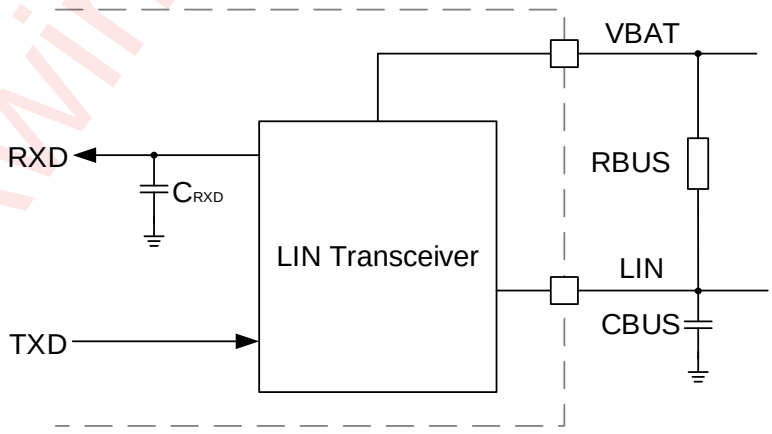


Figure 6 LIN test circuit

SYSTEM AND FLASH OVERVIEW

This chapter describes how to power on, power off, initialize and fine-tune the system, and enter and exit the low-power mode.

POWER-ON

The car battery is powered by VBAT, and there are multiple LDOs inside the chip to establish the on-chip power system:

- LDO3V3: Input: 13.5V, Output: 3.3V
- LDO1V5: Input: 3V, Output: 1.5V

After the power supply is connected to the VBAT pin (the peripheral circuit of AW32F020QNR-Q1 has been confirmed to be intact), the voltage on the chip power pin VBAT pin meets the working voltage range, and the system will start to work.

The following figure shows the power-on timing relationship of AW32F020QNR-Q1 (VBAT, LDO, GPIO, LIN, etc.) :

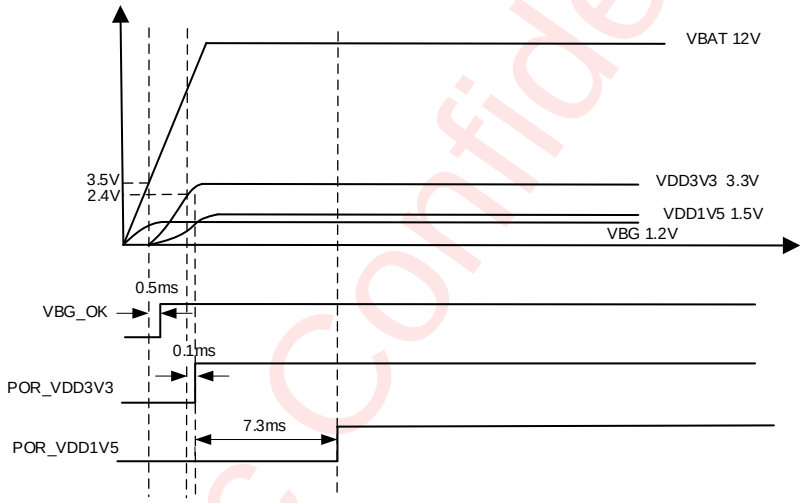


Figure 7 AW32F020QNR-Q1 Power on Timing

After rising from POR_VDD1V5, the hardware initialization time is 2.1ms. Considering the hardware initialization time, it is about 10ms from the start of power on to the chip being able to work normally.

POWER-DOWN

When VDD_3V3 or VBAT is below the preset voltage threshold, the whole chip will be reset immediately.

MEMORY

AW32F020QNR-Q1 has a built-in 16KB SRAM, which can support byte (8 bits), half-word (16 bits), or word (32 bits) read and write access. The built-in FLASH size is 64KB, supports read, write, erase operations, and supports access with 32-bit words. FLASH is used to save the chip execution code, including initialization procedures and various detection, judgment, processing procedures.

- 64KB embedded FLASH for storing programs and data
- 6KB NVR area for storing user configuration information

MEMORY MAPPINGS

Address	Interface	Description
0x00000000 – 0x0000FFFF	FLASH	64Kbytes of Flash Memory, user programmable
0x00010000 – 0x000117FF	NVR	6KB
0x00012000 – 0x1FFFFFFF	NA	-
0x20000000 – 0x20003FFF	SRAM	16KB SRAM
0x20004000 – 0x3FFFFFFF	NA	-
0x40004000 – 0x400040FF	WDT	Watchdog timer
0x40004100 – 0x400041FF	TIMER1	Basic timer 1
0x40004200 – 0x400042FF	TIMER2	Basic purpose timer 2
0x40004300 – 0x400043FF	TIMER3	General purpose timer 3
0x40004400 – 0x400044FF	RCCM	Timer, WDT Clock Reset Controller
0x40004500 – 0x400045FF	FMC	FLASH controller
0x40004600 – 0x400046FF	SYSCTRL	
0x40004700 – 0x400047FF	NA	
0x40004800 – 0x400048FF	GPIOM	MCU GPIO bit control and configuration
0x40004900 – 0x400049FF	SPI	SPI Register
0x40004A00 – 0x40004AFF	I^2C	I^2C Register
0x40004B00 – 0x40004BFF	UART	UART Register
0x40004C00 – 0x40004CFF	NA	
0x40004D00 – 0x4FFFFFFF	NA	-
0x50000000 – 0x500000FF	RCC	Clock & Reset Generator
0x50000100 – 0x500001FF	PMU	Power Management Unit
0x50000200 – 0x500002FF	NA	-
0x50000300 – 0x500003FF	WWDG	Window Watchdog Register
0x50000400 – 0x500004FF	LS_CTRL	LS Controllers
0x50000500 – 0x500005FF	LINS	LIN slave control and PHY registers
0x50000600 – 0x500006FF	LINM	LIN master PHY register
0x50000700 – 0x500007FF	ADC_CTRL	ADC controller
0x50000800 – 0x500008FF	AFIO	Alternate function
0x50000900 – 0x500009FF	SYSCTRL	System configuration and retention memory
0x50000A00 – 0x50000AFF	GPIO	GPIO bit control and configuration
0x50000E00 – 0x50000EFF	WAKEUP TIMER	Wake up timer

DETAILED FUNCTIONAL DESCRIPTION

CORE PERFORMANCE

AW32F020QNR-Q1 integrates a 32-bit CPU core. The platform provides excellent computing performance and efficient interrupt system, while the peripheral devices use less pins and lower power consumption.

A nested vector interrupt controller is built in. The interrupt controller can handle up to 32 maskable interrupt channels and supports four different interrupt priorities.

INTERRUPT AND EXCEPTION VECTORS

The vector table of AW32F020QNR-Q1 is as follows.

Location	Priority	Priority type	Abbreviations
-	-3	Fixed	Reset_Handler
-	-2	Fixed	NMI_Handler
-	-1	Fixed	HardFault_Handler
-	0	Configurable	MemManage_Handler
-	1	Configurable	BusFault_Handler
	2	Configurable	UsageFault_Handler
-	3	Configurable	SVC_Handler
	4	Configurable	DebugMon_Handler
-	5	Configurable	PendSV_Handler
-	6	Configurable	SysTick_Handler
0	7	Configurable	WDT_Handler
1	8	Configurable	Timer1_Handler
2	9	Configurable	Timer2_Handler
3	10	Configurable	Timer3Handler
8	15	Configurable	GPIOM_Handler
9	16	Configurable	SPI_Handler
10	17	Configurable	I^2C_Handler
11	18	Configurable	UART_Handler
16	23	Configurable	LINS_WU_Handler
17	24	Configurable	LINS_PHY_Handler
18	25	Configurable	TIMER_WU_Handler
19	26	Configurable	BOR_Handler
20	27	Configurable	WWDG_Handler
21	28	Configurable	UV_Handler
22	29	Configurable	OV_Handler
23	30	Configurable	LINS_Handler
24	31	Configurable	ADC_Handler
27	34	Configurable	GPIO_Handler
29	36	Configurable	OVTEMP_Handler

FLASH_CTRL

OVERVIEW

AW32F020QNR-Q1 is equipped with a built-in 64KB FLASH memory, which serves as storage for the application program. Upon powering up the chip, the CPU fetches instructions from the FLASH to execute the application program.

FEATURES

- 64KB FLASH main storage area and 6KB user NVR area, 512 bytes per sector
- Support FLASH writing in word
- Support FLASH sector erase, FLASH chip erase
- Support high-speed access to the FLASH storage area using the AHB bus
- Support code protect

FUNCTIONAL DESCRIPTION

FLASH Structure

FLASH area	Name	FLASH Adress	Size (byte)
Main FLASH Memory(64K)	Sector 0	0x0000 0000 - 0x0000 01FF	0.5 Kbyte
	Sector 1	0x0000 0200 - 0x0000 03FF	0.5 Kbyte
	Sector 2	0x0000 0400 - 0x0000 05FF	0.5 Kbyte
	Sector 3	0x0000 0600 - 0x0000 07FF	0.5 Kbyte
	. . .	. . .	. . .
	Sector 124	0x0000 F800 - 0x0000 F9FF	0.5 Kbyte
	Sector 125	0x0000 FA00 - 0x0000 FBFF	0.5 Kbyte
	Sector 126	0x0000 FC00 - 0x0000 FDFF	0.5 Kbyte
	Sector 127	0x0000 FE00 - 0x0000 FFFF	0.5 Kbyte
NVR block (6K)	Usr nvr Sector0	0x0001 0000 - 0x0001 01FF	0.5 Kbyte
	Usr nvr Sector1	0x0001 0200 - 0x0001 03FF	0.5 Kbyte
	Usr nvr Sector2	0x0001 0400 - 0x0001 05FF	0.5 Kbyte
	Usr nvr Sector3	0x0001 0600 - 0x0001 07FF	0.5 Kbyte
	. . .	. . .	. . .
	Usr nvr Sector9	0x0001 1200 - 0x0001 11FF	0.5 Kbyte
	Usr nvr Sector10	0x0001 1400 - 0x0001 13FF	0.5 Kbyte
	Usr nvr Sector11	0x0001 1600 - 0x0001 17FF	0.5 Kbyte

FLASH Read Protection

When the chip is powered on or the system is reset, the debug interface is disabled during the instruction

period of the first 8192 cycles of CPU execution.

Write 0x91827364 to the register FLS_CODE_PROT to enable FLASH read protection and disable the debug interface.

Note: If the read protection function is turned on within 8192 instruction cycles after the power reset or system reset of the chip, the chip will not be connected through the debug interface later.

FLASH Power down Protection

In order to prevent damage to FLASH caused by unexpected power down in the process of FLASH erasing and writing, FLASH power down protection function can be enabled before FLASH erasing and writing operations. Here's how:

Step 1: Start the unexpected power down protection circuit (write 0x1 to register FLS_BOR_CFG);

Step 2: Wait for 100μs;

Step 3: Enable the unexpected power down protection function (write 0x3 to register FLS_BOR_CFG);

FLASH Erase Initialization

FLASH erase and write operations have different timing requirements under different system clock frequencies. The timing parameters of FLASH erase and write operations are configured as follows:

Register	Programming			Sector Erase			Chip Erase		
	16M	8M	4M	16M	8M	4M	16M	8M	4M
T_NVS	0x190	0xC8	0x64	0x190	0xC8	0x64	0x5A0	0x2D0	0x168
T_PGS	0x3C0	0x1E0	0xF0	0x3C0	0x1E0	0xF0	0x3C0	0x1E0	0xF0
T_PROG	0x5C	0x2E	0x17	0x5C	0x2E	0x17	0x5C	0x2E	0x17
T_RCV	0x3C0	0x1E0	0xF0	0x3C0	0x1E0	0xF0	0xDC0	0x6E0	0x370
T_RW	0xC	0x6	0x3	0xC	0x6	0x3	0xC0	0x60	0x30
T_ERASE				0x9600	0x4B00	0x2580	0x84D00	0x42680	0x21340
T_WAKEUP	0x192	0xC8	0x64	0x192	0xC8	0x64	0x192	0xC8	0x64
T_ADSH	0xC	0x6	0x3	0xC	0x6	0x3	0xC	0x6	0x3

FLASH Sector Erase

1) Configure the timing parameters of the FLASH sector erase operation based on the system clock frequency, and reference to the recommended parameters in FLASH erase initialization.

2) Configure the ISP_ADR register: writing the address of the FLASH sector to be erased.

3) Configure the ISP_CMD register: setting the ISP_CMD bit field with the erase instruction. If the target FLASH address is in NVR region, set ISP_NVR bit field to 1, otherwise, set it to 0. Set ISP_CHIP bit field to 0.

4) Configure the ISP_GO register: setting the ISP_GO bit field to 1 to initiate sector erase operation. When this bit is cleared, indicating completion of sector erase operation.

FLASH Chip Erase

1) Configure the timing parameters of the FLASH full-chip erase operation based on the system clock frequency, and reference to the recommended parameters in the FLASH erase initialization.

2) Configure the ISP_CMD register: writing the erase instruction to the ISP_CMD bit field; set ISP_NVR bit field to 0; set ISP_CHIP bit field to 1.

3) Configure the ISP_GO register: writing 1 to the ISP_GO bit field to start the full-chip erase operation; When this bit is cleared, indicating that the full-chip erase operation is complete.

FLASH Write

Users can write data to the address by accessing it directly.

FLASH Read

Users can read FLASH data at a specified address through direct address access.

REGISTER LIST

Register Mapping

FLASH_CTRL_BASE : 0x40004500

Offset	Register Name	Description	Reset Value
0x00	ISP_CR	ISP Control Register	0x100
0x04	ISP_ADDR	ISP Address Register	0x0
0x10	ISP_CMD	ISP Command Register	0x20
0x14	ISP_GO	ISP Go Register	0x0
0x20	T_NVS	T_NVS Register	0x190
0x24	T_PGS	T_PGS Register	0x3C0
0x28	T_PROG	T_PROG Register	0x5C
0x2C	T_RCV	T_RCV Register	0x3C0
0x30	T_RW	T_RW Register	0xC
0x34	T_ERASE	T_ERASE Register	0x9600
0x38	T_WAKEUP	T_WAKEUP Register	0x192
0x40	T_ADSH	T_ADSH Register	0xC
0x48	FLS_CODE_PROT	FLASH Code Protection Register	0x0
0x4C	BOR_CFG	FLASH BOR Configue Register	0x0
0x54	BOR_STATE	BOR State Register	0x0

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write

ISP_CR: Address(0x000)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RO	Not used	0
8	BYPASS_WR_FF	RW	0: Writing FF to flash will not be skipped 1: Writing FF to flash will be skipped	1
7:0	Reserved	RO	Not used	0

ISP_ADR: Address(0x004)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	Reserved	RO	Not used	0
15:0	ISP_ADR	RW	Indicates the destination address of the ISP operation: 0x0000~0xFFFF	0

ISP_CMD: Address(0x010)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RO	Not used	0
8	READONLY	RW	READONLY=1 to prohibit programming and erase operations.	0
7:6	Reserved	RO	Not used	0
5	ISP_NVR	RW	NVR sector selection. 0: Select FLASH main area; 1: Select NVR area .	0
4	ISP_CHIP	RW	Chip selection signal. 0: sector erase; 1: chip erase.	0
3:0	ISP_CMD	RW	ISP command. 0101: Erase ; Others: Reserved.	0

ISP_GO: Address(0x014)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:1	Reserved	RO	Not used	0
0	ISP_GO	RW	ISP operation start signal.(software write 1, hardware reset) 0: ISP operation completed, when ISP_GO is zero, software can be allowed to configure registers; 1: ISP operation is in progress.	0

T_NVS: Address(0x020)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:11	Reserved	RO	Not used	0
10:0	T_NVS	RW	PROG/ERASE/CEb/NVR/Address set up time to Web.(unit: system clock period) Program: Min=20μs; 16MHz: T_NVS=0x190; 8MHz: T_NVS=0xC8; 4MHz: T_NVS=0x64. Sector erase: Min=20μs; 16MHz: T_NVS=0x190; 8MHz: T_NVS=0xC8; 4MHz: T_NVS=0x64. Chip Erase: Min=80μs; 16MHz: T_NVS=0x5A0; 8MHz: T_NVS=0x2D0; 4MHz: T_NVS=0x168.	0x190

T_PGS: Address(0x024)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:11	Reserved	RO	Not used	0

10:0	T_PGS	RW	Web low level to PROG2 high level to set up tiime.(unit: system clock period) T_PGS: Min=50 μs ,Max=70 μs. 16MHz: T_PGS=0x3C0; 8MHz: T_PGS=0x1E0; 4MHz: T_PGS=0xF0.	0x3C0

T_PROG: Address(0x028)

Bit	Symbol	R/W	Description	Default
31:7	Reserved	RO	Not used	0
6:0	T_PROG	RW	Byte program time.(unit: sysytem clock period) T_PROG Min=5μs, max=6.5μs. 16MHz: T_PROG=0x5C; 8MHz: T_PROG=0x2E; 4MHz: T_PROG=0x17.	0x5C

T_RCV: Address(0x02C)

Bit	Symbol	R/W	Description	Default
31:13	Reserved	RO	Not used	0
12:0	T_RCV	RW	program/erase reset time(unit: sysytem period) Program: Min=50μs; 16MHz: T_RCV=0x3C0; 8MHz: T_RCV=0x1E0; 4MHz: T_RCV=0xF0. Sector erase: Min=50μs; 16MHz: T_RCV=0x3C0; 8MHz: T_RCV=0x1E0; 4MHz: T_RCV=0xF0. Chip Erase: Min=200 μs; 16MHz: T_RCV=0xDC0; 8MHz: T_RCV=0x6E0; 4MHz: T_RCV=0x370.	0x3C0

T_RW: Address(0x030)

Bit	Symbol	R/W	Description	Default
31:8	Reserved	RO	Not used	0
7:0	T_RW	RW	PROG/ERASE signal low level to next operation delay time.(unit: sysytem time) Program: Min=0.5μs; 16MHz: T_RW=0xC; 8MHz: T_RW=0x6; 4MHz: T_RW=0x3. Sector erase: Min=0.5μs; 16MHz: T_RW=0xC; 8MHz: T_RW=0x6; 4MHz: T_RW=0x3. Chip Erase: Min=10μs; 16MHz: T_RW=0xC0; 8MHz: T_RW=0x60; 4MHz: T_RW=0x30.	0xC

T_ERASE: Address(0x034)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:20	Reserved	RO	Not used	0
19:0	T_ERASE	RW	ISP erase time.(unit : sysytem clock period) Sector erase: min=2ms, max=3ms; 16MHz: T_ERASE=0x9600; 8MHz: T_ERASE=0x4B00; 4MHz: T_ERASE=0x2580. Chip erase: min=30ms, max=40ms; 16MHz: T_ERASE=0x84D00; 8MHz: T_ERASE=0x42680; 4MHz: T_ERASE=0x21340.	0x9600

T_WAKEUP: Address(0x038)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RO	Not used	0
8:0	T_WAKEUP	RW	Flash wakeup time(unit: sysytem period) FWUP=1: T_WAKEUP min = 10μs; FWUP=0: T_WAKEUP min = 20μs; 16MHz : T_WAKEUP=0x192; 8MHz: T_WAKEUP=0xC8; 4MHz: T_WAKEUP=0x64.	0x192

T_ADSH: Address(0x040)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:5	Reserved	RO	Not used	0
4:0	T_ADSH	RW	BYTE[3:0]/Address/data setup/hold timing set. T_ADS\T_ADH: Min=0.5μs; 16MHz :T_ADSH=0xC; 8MHz: T_ADSH=0x6; 4MHz: T_ADSH=0x3.	0xC

FLS_CODE_PROT: Address(0x048)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:0	FLS_CODE_PROT	RW	Write 0x91827364 to enable flash read protection function. 0: The flash read protection function is enabled; 1: The flash read protection function is disabled	0

BOR_CFG: Address(0x04C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:2	Reserved	RO	Not used	0
1	FBOR_BOR_VALID	RW	This register is valid when FBOR_EN is set to 1. 0: Disable sudden power failure protection function 1: Enable sudden power failure protection function	0
0	FBOR_EN	RW	0: Disable power down protection circuit 1: Enable power down protection circuit	0

BOR_STATE: Address(0x054)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:1	Reserved	RO	Not used	0
0	FBOR_BOR	RW	Unexpected power failure indicator, this bit is written 1 to clear 0 0: No unexpected power failure 1: Unexpected power failure	0

TIMER1/ TIMER2

OVERVIEW

TIMER1 and TIMER2 both contain a 32-bit counter that can be used as basic timers to provide timing functionality to the system.

FEATURES

- 32-bit counter, working in the system clock domain.
- Support interrupt control functions, including interrupt generation, interrupt clearing, and interrupt masking.
- When the timer is working, TimerLoadCount changes. It will only be updated by the timer when it reaches 0 or when it is restarted. Otherwise, the timer will continue to decrease until it reaches 0 before updating TimerLoadCount.

FUNCTIONAL DESCRIPTION

1) Start the TIMER Clock

The enable and disable of TIMER1 and TIMER2 clocks can be achieved through RCCM.CLKENR register.

2) Configure TIMER loading value

The TIMER reload value can be configured through the TIMERx.TimerLoaderCount register.

3) Configure TIMER interrupt

TIMER interrupts are enabled and masked by the TIMERx.TimerControlReg register.

4) Start TIMER interrupt

Enable the TIMER global interrupt and configure the appropriate interrupt priority according to the application.

5) Start the TIMER

TIMER enable and disable can be configured by the TIMERx.TimerControlReg register.

6) TIMER interrupt generating

When the TIMER is enabled, the counter starts counting down from the value configured in the TIMERx.TimerLoaderCount register. When the count is decreased to 0, the TIMER interrupt is generated, and the counter value is restored to the value configured in the TIMERx.TimerLoaderCount register, and the counting down starts again.

7) TIMER interrupt clearing

After the TIMER interrupt is generated, the register TIMERx.TimerEOI needs to be written to clear the TIMER interrupt state.

8) Reconfigure the TIMER count value

During the TIMER counting process, the register TIMERx.TimerLoadCount can be configured at any time to update the load count value. At this time, the original counting process is not interrupted. When the original counting process is decremented to 0, the updated load count value is returned to continue decrementing.

9) Disable the TIMER

The TIMERx.TimerControlReg register can be configured to disable the TIMER at any time during TIMER working.

USAGE RESTRICTIONS

- TimerLoadCount must be set before starting timer_en. Timer must be disabled and then enabled again if TimerLoadCount is modified.
- The TimerLoadCount time interval cannot be set to 0x1~0xF, which is a relatively small value.The set time should be more than 1ms, TimerLoadCount>0x3FF.

REGISTER LIST

Register Mapping

TIMER1_BASE : 0x40004100

TIMER2_BASE : 0x40004200

TIMER register:

Offset	Register Name	Description	Reset Value
0x00	TimerLoadCount	Counter.	0xFFFFFFFF
0x08	TimerControlReg	Control.	0
0x0C	TimerEOI	Write 1 Clear Interrupt Status.	0
0x10	TimerIntStat	Interrupt status.	0

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write, WC: Write clear

TimerLoadCount: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:0	TimerLoadCount	RW	Timer loading value, which is loaded automatically when the Timer starts or when the count decrements to 0 in user-defined mode.	0xFFFFFFFF

TimerControlReg: Address(0x08)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:3	Reserved	RO	Not used	0
2	TimerIntMsk	RW	Timer interrupt mask 0: Disable 1: Enable	0
1	Reserved	RO	Not used	0
0	TimerEn	RW	Timer enable 0: Disabled 1: Enable	0

TimerEOI: Address(0x0C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:1	Reserved	RO	Not used	0
0	TimerIntclr	WC	Write 1 Clear Interrupt Status. 1: Enable	0

TimerIntStat: Address(0x10)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:1	Reserved	RO	Not used	0
0	TimerIntstate	RO	Timer interrupt status register 1: There is an interrupt generated 0: There is no interrupt generated	0

TIMER3

OVERVIEW

The general-purpose timer TIMER3 consists of a 16-bit auto-reload counter driven by a programmable prescaler. It can be used to generate output waveforms (output compare and PWM). The pulse length and waveform period can be modulated from a few microseconds to a few milliseconds using the timer prescaler.

FEATURES

- Working clock: 16 MHz
- 16-bit counter, counter direction configurable as up, down, or up-down
- Supports clock prescaling, prescaler coefficient configuration range: 1 to 65535
- A total of 4 channels, supports output comparison and PWM generation
- Supports interrupt generation for the following events:
  - Update: counter overflow/underflow, counter initialization (via software)
  - Output comparison

FUNCTIONAL DESCRIPTION

Time base unit

The main components of a universal timer are a 16bit counter and associated auto-reload registers that support up, down, or up-down counting modes. The clock unit of the counter is generated by the master clock via a pre-divider. Time base units include:

- Counter register (TIM_CNT)
- Pre-division register (TIM_PSC)
- Auto reload register (TIM_ARR)

The pre-division register can divide the master clock by any value between 1 and 65536, and the pre-division register has a buffer, that is, a shadow register. User modifications to the pre-split register are passed into the shadow register at each update event.

The auto-reload register also has a shadow register, and the user's access to the auto-reload register directly manipulates the auto-reload register. Depending on the auto-reload enable bit setting, the contents of the auto-reload register will be transferred to the shadow register either immediately or at each update event.

Counter modes

The universal timer supports three count modes, namely, up count, down count and up and down two-way count. To switch the count mode, you need to turn off the counter enable, then switch the mode, and finally turn on the counter enable.

- **Up counting mode**

In count up mode, the counter counts from 0 to the autoload value, then counts from 0 and produces a counter overflow. If a repeat counter is used, an update event is generated when the upward count reaches the set number of repeat counts, otherwise an update event is generated every time the counter overflows.

- **Down counting mode**

In count down mode, the counter counts from TIM_ARR to 0, then counts from the new TIM_ARR and produces a counter overflow. If a repeat counter is used, an update event is generated when the upward count reaches the set number of repeat counts, otherwise an update event is generated every time the counter overflows.

- **Central count mode(Up/Down count mode)**

In two-way counting mode, counters count from 0 to TIM_ARR-1, producing a counter overflow event, then count down to 1, producing a counter overflow event, and then count again from 0. In this mode, the counter direction control bit DIR, which is used to indicate the current count direction, cannot be written.

When an update event occurs, all registers are updated and the hardware sets the update flag bit.

- Autoload the shadow register passing in the TIM_ARR value
- The pre-split shadow register is passed the value of TIM_PSC

Force output

In output mode (CCxS bit = 00 in the TIMx_CCMRx register), each output comparison signal (OCxREF, then OCx) can be directly forced to a valid or invalid level by software, independent of any comparison between the output comparison register and the counter.

To force the output comparison signal (OCxREF/OCx) to a valid level, you simply write 101 in the OCxM bit of the corresponding TIMx_CCMRx register. Therefore, OCxREF is forced to a high level (OCxREF is always active at a high level) and OCx acquires a value opposite to the polarity of CCxP.

For example, CCxP=0 (the OCx high level is valid) => OCx is forced to be high.

The OCxREF signal can be forced to a low level by writing the OCxM bit in the TIMx_CCMRx register to 100.

In any case, the comparison between TIMx_CCRx shadow registers and counters is still performed and flags are allowed to be set.

Output compare

This function is used to control the output waveform or indicate when a time period has passed.

In output comparison mode, updating event UEV has no effect on OCxREF and OCx output. The timing resolution is a count of the counter. The output comparison mode can also be used to output a single Pulse (in One Pulse mode).

Configuration process:

1. Select the counter clock (internal, external, pre-divider).
2. Write the required data in the TIMx_ARR and TIMx_CCRx registers.
3. Set the CCxIE bit if you want to generate an interrupt.
4. Select an output mode. For example:

- When CNT matches CCRx, write OCxM = 011 to switch OCx output pins
- Write OCxPE = 0 to disable the preload register
- Write CCxP = 0 to select the high level active polarity
- Write CCxE = 1 to enable output

5. Enable the counter by setting the CEN bit in the TIMx_CR1 register.

The TIMx_CCRx register can be updated at any time through software to control the output waveform, provided that the pre-loaded register is not enabled (OCxPE='0', otherwise the TIMx_CCRx shadow register is only updated at the next update event UEV).

PWM output

The pulse width modulation mode allows you to generate a signal whose frequency is determined by the value of the TIMx_ARR register and whose duty cycle is determined by the value of the TIMx_CCRx register.

By writing "110" (PWM mode 1) or "111" (PWM mode 2) in the OCxM bit of the TIMx_CCMRx register, the PWM mode can be selected independently on each channel (one PWM output per OCx). You must enable the corresponding preload register by setting the OCxPE bit in the TIMx_CCMRx register, and eventually enable the auto-reload preload register by setting the ARPE bit in the TIMx_CR1 register.

Since the preloaded registers are only transferred to the shadow registers when an update event occurs, you must initialize all registers by setting the UG bit in the TIMx_EGR register before starting the counter.

OCx polarity can be programmed by software using the CCxP bit in the TIMx_CCER register. It can be programmed to be high or low level active. OCx output is enabled by the CCxE bit in the TIMx_CCER register.

In PWM mode (1 or 2), always compare TIMx_CNT and TIMx_CCRx to determine whether TIMx_CCRx ≤ TIMx_CNT or TIMx_CNT ≤ TIMx_CCRx (depending on the direction of the counter).

Depending on the CMS bit in the TIMx_CR1 register, the timer is capable of generating PWM in either edge-aligned mode or centre-aligned mode.

- **PWM edge alignment mode - up counting mode**

When the DIR bit in the TIMx_CR1 register is 0, the up count is valid.

As long as TIMx_CNT < TIMx_CCRx, the reference PWM signal OCxREF is high, otherwise it becomes low. If the comparison value in TIMx_CCRx is greater than the automatically overloaded value (in TIMx_ARR), OCxREF remains 1. If the comparison value is 0, OCxREF remains 0.

- **PWM edge alignment mode - down counting mode**

When the DIR bit in the TIMx_CR1 register is 1, the down count is valid.

As long as TIMx_CNT > TIMx_CCRx, the reference signal OCxREF is low, otherwise it becomes high. If the comparison value in TIMx_CCRx is greater than the automatically overloaded value in TIMx_ARR, OCxREF remains 1. In this mode, 0% PWM is not possible.

- **PWM central alignment mode**

Center alignment mode is active when the CMS bit in the TIMx_CR1 register is different from 00 (all remaining configurations have the same effect on the OCxREF/OCx signal). Depending on the CMS bit configuration, the comparison flag is set when the counter counts up, down, or up and down. The direction bit (DIR) in the

TIMx_CR1 register is updated by hardware and cannot be changed by software.

Tips for using Center alignment mode:

• When starting in center alignment mode, use the current up and down configuration. This means that the counter increments or deciles the count depending on the value written to the DIR bit in the TIMx_CR1 register. In addition, the software must not change DIR and CMS bits at the same time.

• It is not recommended to write counters when running in center alignment mode, as this can lead to unexpected results. In particular:

- If the value written in the counter is greater than the automatic overload value (TIMx_CNT>TIMx_ARR), the direction is not updated. For example, if the counter is counting up, it continues to count up.

- If 0 is written in the counter or the TIMx_ARR value is written but no update event UEV is generated, the direction is updated.

• The safest way to use the center alignment mode is to generate an update (set the UG bit in the TIMx_EGR register) by software before starting the counter, rather than writing the counter while it is running.

REGISTER LIST

Register Mapping

TIMER3_BASE : 0x40004300

TIMER register:

Offset	Register Name	Description	Reset Value
0x00	TIM_CR1	Control Register 1	0x0
0x0C	TIM_DIER	Interrupt Enable Register	0x0
0x10	TIM_SR	Status register	0x0
0x14	TIM_EGR	Event Generation Register	0x0
0x18	TIM_CCMR1	Compare Register 1	0x0
0x1C	TIM_CCMR2	Compare Register 2	0x0
0x20	TIM_CCER	Comparison Enable Register	0x0
0x24	TIM_CNT	Counter register	0x0
0x28	TIM_PSC	Prescaler register	0x0
0x2C	TIM_ARR	Automatic reload register	0xffff
0x34	TIM_CCR1	Comparison Value Register 1	0x0
0x38	TIM_CCR2	Comparison Value Register 2	0x0
0x3C	TIM_CCR3	Comparison Value Register 3	0x0
0x40	TIM_CCR4	Comparison Value Register 4	0x0

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write, WC: Write clear

TIMx_CR1: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RW	Not used	0
7	ARPE	RW	0: TIMx_ARR register is not buffered 1: TIMx_ARR register is buffered	0

6:5	CMS	RW	00: Edge-aligned mode. The counter counts up or down according to the direction bit (DIR). 01: Center-aligned mode 1. The counter alternates between counting up and down. The output compare interrupt flag of the channel is set only when the counter counts down. 10: Center-aligned mode 2. The counter alternates between counting up and down. The output compare interrupt flag of the channel is set only when the counter counts up. 11: Center-aligned mode 3. The counter alternates between counting up and down. The output compare interrupt flag of the channel is set when the counter counts up or down. Note: It is not allowed to switch from edge-aligned mode to center-aligned mode as long as the counter is enabled (CEN = 1).	0
4	DIR	RW	0: The counter is used as an up counter. 1: The counter is used as a down counter. Note: This bit is read-only when the timer is configured in center-aligned mode.	0
3	Reserved	RW	Not used	0
2	URS	RW	This bit is set and cleared by software to select the UEV event source. 0: If enabled, any of the following events will generate an update interrupt. - Counter overflow/underflow - Setting the UG bit 1: If enabled, only counter overflow/underflow will generate an update interrupt.	0
1	UDIS	RW	This bit is set and cleared by software to enable/disable UEV event generation. 0: UEV enabled. An update (UEV) event is generated by one of the following events: - Counter overflow/underflow - Setting the UG bit 1: UEV disabled. No update event is generated and the shadow registers keep their values (ARR, PSC, CCRx). However, if the UG bit is set, the counter and prescaler are reinitialized.	0
0	CEN	RW	0: Counter disabled 1: Counter enabled Note: The external clock, gated mode, and encoder mode can only function after the CEN bit is set by software. However, the trigger mode can automatically set the CEN bit by hardware.	0

TIMx_DIER: Address(0x0C)

Bit	Symbol	R/W	Description	Default
31:5	Reserved	RW	Not used	0
4	CC4IE	RW	Allow capture/compare 4 interrupt 0: Disable capture/compare 4 interrupt 1: Enable capture/compare 4 interrupt	0
3	CC3IE	RW	Allow capture/compare 3 interrupt 0: Disable capture/compare 3 interrupt 1: Enable capture/compare 3 interrupt	0
2	CC2IE	RW	Allow capture/compare 2 interrupt 0: Disable capture/compare 2 interrupt 1: Enable capture/compare 2 interrupt	0
1	CC1IE	RW	Allow capture/compare 1 interrupt 0: Disable capture/compare 1 interrupt 1: Enable capture/compare 1 interrupt	0
0	UIE	RW	Allow update interrupts 0: Prohibit update interrupts 1: Allow update interrupts	0

TIMx_SR: Address(0x10)

Bit	Symbol	R/W	Description	Default
31:5	Reserved	RC_W0	Not used	0
4	CC4IF	RC_W0	Capture/Compare 4 Interrupt Flag Refer to the description of CC1IF	0
3	CC3IF	RC_W0	Capture/Compare 3 Interrupt Flag Refer to the description of CC1IF	0
2	CC2IF	RC_W0	Capture/Compare 2 Interrupt Flag Refer to the description of CC1IF	0
1	CC1IF	RC_W0	Compare 1 Match Flag This bit is set by hardware when the counter value matches the compare value, except in Center-aligned mode (refer to the CMS bit in the TIMx_CR1 register). It is cleared by software. 0: No match occurs; 1: The value of TIMx_CNT matches the value of TIMx_CCR1. When the content of TIMx_CCR1 is greater than the content of TIMx_ARR, the CC1IF bit goes high on counter overflow (in Up and Up/Down counting modes) or underflow (in Down counting mode).	0
0	UIF	RC_W0	Update interrupt flag This bit is set by hardware when an update event occurs and cleared by software. 0: No update event has occurred; 1: An update event is pending. This bit is set by hardware when the register is updated:	0

			- If TIMx_CR1 register's UDIS = 0, an update event occurs when the counter overflows or underflows (counter overflow or underflow); - If TIMx_CR1 register's UDIS = 0 and URS = 0, an update event occurs when TIMx_EGR register's UG = 1 (software reinitializes CNT).	

TIMx_EGR: Address(0x14)

Bit	Symbol	R/W	Description	Default
31:5	Reserved	W	Not used	0
4	CC4G	W	Generate a comparison of 4 events Refer to the description of CC1G	0
3	CC3G	W	Generate a comparison of 3 events Refer to the description of CC1G	0
2	CC2G	W	Generate a comparison of 2 events Refer to the description of CC1G	0
1	CC1G	W	Generate a comparison event 1 This bit is set to 1 by software to generate a comparison event, and is automatically cleared to 0 by hardware. 0: No action; 1: Generate a comparison event on channel CC1: Set CC1IF = 1, and if the corresponding interrupt is enabled, generate the corresponding interrupt.	0
0	UG	W	Generate update event This bit is set by software and cleared by hardware automatically. 0: No action; 1: Reinitialize the counter and generate an update event. Note that the counter of the prescaler is also cleared (but the prescaler coefficient remains unchanged). If in center-aligned mode or DIR = 0 (up counting), the counter is cleared to 0. If DIR = 1 (down counting), the counter takes the value of TIMx_ARR.	0

TIMx_CCMR1: Address(0x18)

Bit	Symbol	R/W	Description	Default
31:15	Reserved	RW	Not used	0
14:12	OC2M	RW	Output Comparison Mode 2 Refer to the description of OC1M	0
11	OC2PE	RW	Output Compare 2 Preload Enable Refer to OC1PE description	0
10:7	Reserved	RW	Not used	0
6:4	OC1M	RW	Output Compare 1 Mode This bit defines the action of the output reference signal	0

			OC1REF, which determines the value of OC1. OC1REF is active high, while the effective level of OC1 depends on the CC1P bit. 000: Frozen. The comparison between the output compare register TIMx_CCR1 and the counter TIMx_CNT has no effect on OC1REF (this mode is used to generate a timing reference). 001: Set channel 1 to the active level on match. When the value of the counter TIMx_CNT is equal to the value of the compare register 1 (TIMx_CCR1), force OC1REF high. 010: Set channel 1 to the inactive level on match. When the value of the counter TIMx_CNT is equal to the value of the compare register 1 (TIMx_CCR1), force OC1REF low. 011: Toggle. When TIMx_CCR1 = TIMx_CNT, toggle the level of OC1REF. 100: Force inactive level. Force OC1REF low. 101: Force active level. Force OC1REF high. 110: PWM mode 1 - In the up-counting direction, channel 1 is active when TIMx_CNT < TIMx_CCR1, otherwise it is inactive; in the down-counting direction, channel 1 is inactive when TIMx_CNT > TIMx_CCR1 (OC1REF = 0), otherwise it is active (OC1REF = 1). 111: PWM mode 2 - In the up-counting direction, channel 1 is inactive when TIMx_CNT < TIMx_CCR1, otherwise it is active; in the down-counting direction, channel 1 is active when TIMx_CNT > TIMx_CCR1, otherwise it is inactive. Note 1: In PWM mode 1 or PWM mode 2, the level of OC1REF changes only when the comparison result changes or when switching from the frozen mode to the PWM mode in the output compare mode.	
3	OC1PE	RW	Output Compare 1 Preload Enable 0: Disable the preload function of the TIMx_CCR1 register. The TIMx_CCR1 register can be written to at any time, and the new value takes effect immediately. 1: Enable the preload function of the TIMx_CCR1 register. Read and write operations only affect the preload register. The preload value of TIMx_CCR1 is loaded into the current register when an update event occurs.	0
2:0	Reserved	RW	Not used	0

TIMx_CCMR2: Address(0x1C)

Bit	Symbol	R/W	Description	Default
31:15	Reserved	RW	Not used	0
14:12	OC4M	RW	Output comparison 4 mode Refer to the description of OC1M	0
11	OC4PE	RW	Output Compare 4 Preload Enable	0

			Refer to OC1PE description	
10:7	Reserved	RW	Not used	0
6:4	OC3M	RW	Output comparison 3 mode Refer to the description of OC1M	0
3	OC3PE	RW	Output Compare 3 Preload Enable Refer to OC1PE description	0
2:0	Reserved	RW	Not used	0

TIMx_CCER: Address(0x20)

Bit	Symbol	R/W	Description	Default
31:14	Reserved	W	Not used	0
13	CC4P	W	Input/Capture 4 Output Polarity Refer to the description of CC1P.	0
12	CC4E	W	Input/Capture 4 Output Enable Refer to the description of CC1E.	0
11:10	Reserved	W	Not used	0
9	CC3P	W	Input/Capture 3 Output Polarity Refer to the description of CC1P.	0
8	CC3E	W	Input/Capture 3 Output Enable Refer to the description of CC1E.	0
7:6	Reserved	W	Not used	0
5	CC2P	W	Input/Capture 2 Output Polarity Refer to the description of CC1P.	0
4	CC2E	W	Input/Capture 2 Output Enable Refer to the description of CC1E.	0
3:2	Reserved	W	Not used	0
1	CC1P	W	Channel 1 output polarity 0: OC1 high level effective 1: OC1 low level effective	0
0	CC1E	W	Channel 1 output enable 0: Off - OC1 output is prohibited. 1: On - OC1 signal is output to the corresponding output pin.	0

TIMx_CNT: Address(0x24)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RW	Not used	0
15:0	CNT	RW	The value of the counter	0

TIMx_PSC: Address(0x28)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RW	Not used	0

15:0	PSC	RW	The value of the prescaler The clock frequency of the counter (CK_CNT) is equal to fCK_PSC / (PSC[15:0] + 1). PSC contains the value to be loaded into the current prescaler register when an update event occurs.	0

TIMx_ARR: Address(0x2C)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RW	Not used	0
15:0	ARR	RW	The value of automatic reloading ARR contains the value that will be loaded into the actual automatic reloading register. For detailed information, refer to the update and action of ARR in the time base unit. When the value of automatic reloading is empty, the counter does not work.	0xffff

TIMx_CCR1: Address(0x34)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RW	Not used	0
15:0	CCR1	RW	The value of Compare 1 is compared. CCR1 contains the value loaded into the current Compare 1 register (the preload value). If the preload feature is not selected in the TIMx_CCMR1 register (OC1PE bit), it is always loaded into the current register. Otherwise, the preload value is only loaded into the current Compare 1 register when an update event occurs. The current Compare register contains the value compared with the counter TIMx_CNT and outputs a signal on the OC1 port.	0

TIMx_CCR2: Address(0x38)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RW	Not used	0
15:0	CCR2	RW	Compare 2 value CCR2 contains the value loaded into the current Compare 2 register (preload value). If the preload feature is not selected in the TIMx_CCMR1 register (OC2PE bit), it is always loaded into the current register. Otherwise, the preload value is loaded into the current Compare 2 register only when an update event occurs. The current compare register contains the value compared with the counter TIMx_CNT and outputs a signal on the OC2 port.	0

TIMx_CCR3: Address(0x3C)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RW	Not used	0
15:0	CCR3	RW	Compare 3 value CCR3 contains the value loaded into the current Compare 3	0

			register (preload value). If the preload feature is not selected in the TIMx_CCMR2 register (OC3PE bit), it is always loaded into the current register. Otherwise, the preload value is loaded into the current Compare 3 register only when an update event occurs. The current compare register contains the value compared with the counter TIMx_CNT and outputs a signal on the OC3 port.	

TIMx_CCR4: Address(0x40)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RW	Not used	0
15:0	CCR4	RW	Compare 4 value The CCR4 contains the value loaded into the current Compare 4 register (preload value). If the preload feature is not selected in the TIMx_CCMR2 register (OC4PE bit), it is always loaded into the current register. Otherwise, the preload value is loaded into the current Compare 4 register only when an update event occurs. The current compare register contains the value compared with the counter TIMx_CNT and outputs a signal on the OC4 port.	0

I²C

OVERVIEW

I²C is a two-wire, bidirectional serial bus that provides a simple and effective method for data communication between devices. The I²C standard is a multi-master bus that includes conflict detection and arbitration mechanisms to prevent data conflicts when two or more masters attempt to control the bus simultaneously.

FEATURES

- Support for both master and slave modes;
- Bidirectional data transmission between master and slave;
- Multi-master bus (without a central master);
- Arbitration for simultaneous data transmission by multiple masters, ensuring that serial data on the bus is not corrupted;
- Support for 7-bit addressing mode and 10-bit addressing mode;
- Support for broadcast addressing;
- Fast mode (up to 400 kHz) and standard mode (up to 100 kHz);
- Programmable SDA hold time.

FUNCTIONAL DESCRIPTION

Mode Selection

I²C supports two operation modes: master and slave. In master mode, if bus arbitration fails, I²C immediately switches to slave mode so that other masters can address the current slave address.

Slave Mode

1) Initialize configuration

   To configure I²C as slave mode, the following steps are required:

1. Disable I²C by writing 0 to the I²C _ENABLE[0] bit.
2. Write the I²C_SAR register to configure the slave address.
3. Write the I²C _CON register to configure the addressing mode. Write 0 to bit 6 to enable I²C slave mode and write 0 to bit 0 to disable the master mode.
4. Enable the I²C module by writing 1 to I²C_ENABLE[0].

2) Slave single-byte transmission operation

When another I²C master device on the bus addresses I²C and requests data, I²C acts as a slave sender and the sending steps are as follows:

1. The other I²C master device initiates an I²C transfer, and the address it sends matches the address in the slave I²C_SAR register.
2. DW_I²C responds to the address sent by the master and identifies the transmission direction, with the slave acting as the sender.
3. DW_I²C generates a read request interrupt RD_REQ and pulls SCL low to hold the I²C bus. This is a waiting state until the software responds to the read request interrupt. If the read request interrupt is masked (IC_INTR_MASK[5] is set to 0), it is recommended that the software periodically query the interrupt status bit to respond to the read request interrupt in a timely manner. When IC_RAW_INTR_STAT[5] is read as 1, it must be considered that RD_REQ has occurred; The interrupt handling function needs to prepare the data to be sent by IIC.
4. If there is already data in the TX FIFO before receiving the read request, I²C generates a transmission abort interrupt (TX_ABRT) to refresh the data in the TX FIFO. Because the TX FIFO enters a refresh and reset state when the abort interrupt occurs, it is necessary to read the I²C _CLR_TX_ABRT register to exit the refresh and reset state of the TX FIFO before writing data to the TX FIFO. If the transmission interrupt is masked (IC_INTR_MASK[6] is set to 0), it is recommended that the software periodically query the interrupt status bit to respond to the read request interrupt in a timely manner. When IC_RAW_INTR_STAT[6] is read as 1, it must be considered that TX_ABRT has occurred;
5. The software writes the data to be sent to the I²C _DATA_CMD register.
6. The software clears the request interrupt and the transmission abort interrupt by reading the I²C _CLR_RD_REQ and IC_CLR_TX_ABRT registers.
7. DW_ I²C starts transmitting data.
8. The master may keep the I²C bus by sending a RESTART condition or release the bus by sending a STOP condition.

3) Single-byte receiving operation from the slave machine

When another I²C master device on the bus addresses I²C and sends data, I²C acts as the slave receiver, and the receiving steps are as follows:

1. Another I²C master device initiates an I²C transmission, and the address sent by the master matches the address in the I²C _SAR register of the slave.
2. I²C responds to the address sent by the master and identifies the transmission direction, with the slave acting as the receiver.
3. I²C receives the data byte sent by the master and saves it in the receive FIFO.
4. I²C generates a receive full interrupt RX_FULL (IC_RAW_INTR_STAT[2]).
5. If the read request interrupt is masked (IC_INTR_MASK[2] is set to 0), it is recommended that the software periodically check the interrupt status bit to respond to the read request interrupt in a timely manner.
6. The software reads the data in I²C _DATA_CMD.
7. The master may keep the I²C bus by sending a RESTART condition or release the bus by sending a STOP condition.
8. If the RX FIFO is full when receiving data, an overflow will occur at this time, and I²C will continue to

receive data. Therefore, the software must detect the overflow signal in a timely manner and take appropriate measures to recover the lost data.

4) Transfer operation from the slave block

1. This IP is designed to handle more data in the TX FIFO, so that subsequent read requests can directly obtain data without waiting for the read request interrupt to be responded to. This eliminates the delay in obtaining data due to the TX FIFO having only one byte of data and the response interrupt being triggered.
2. This mode only occurs when the I²C acts as a slave and sends data. If the external master responds to the data sent by the slave but the slave's TX FIFO is empty, the slave will pull down SCL to hold the I²C bus and generate a read request interrupt, waiting for the CPU to write data into the TX FIFO.
3. If the read request interrupt is masked, it is recommended to periodically read the value of the I²C_RAW_INTR_STAT[5] register to check the status of the read request interrupt.
4. A read request interrupt is generated when there is a read request. Like other interrupts, it must be cleared before exiting the interrupt service routine. In the interrupt service routine, one or more data can be written into the TX FIFO. When sending data to the master, if the master responds to the last byte, the slave must generate a read request interrupt again because the master is requesting more data.
5. If the programmer knows in advance that the master is requesting an N-byte data packet, N bytes of data can be written into the TX FIFO, and the master will process these N bytes of data as a data stream. For example, as long as the master responds to the data sent by the slave and there is still data in the TX FIFO, the slave will continuously send data to the master. Therefore, there is no need to pull down SCL to hold the I²C bus or generate a read request response.
6. If the master is to receive N bytes of data and the TX FIFO of the I²C slave is written with more than N bytes of data, the slave will send N bytes of data, clear the TX FIFO, and ignore any excess data.

Host Mode

1) Initialize configuration

To configure I²C as the master mode, the following steps are required:

1. Write 0 to the I²C _ENABLE[0] bit to disable I²C.
2. Write to the I²C _CON register to configure the communication speed and addressing mode (7-bit or 10-bit addressing). Write 1 to bit 6 to disable I²C slave mode and write 1 to bit 0 to enable the master.
3. Write the address of the slave device to be addressed to the I²C _TAR register.
4. In high-speed mode, write the expected master code to the I²C _HS_MADDR register. The master code is defined by the programmer. This step is ignored in normal mode and fast mode.
5. Write 1 to I²C _ENABLE[0] to enable the I²C module.
6. Write the transfer direction and data to the I²C _DATA_CMD register. If the I²C _DATA_CMD register is written before I²C is enabled, both the data and command will be lost because the register remains cleared before the I²C module is enabled.

2) Host transmission and host reception

I²C supports dynamic switching between read and write operations. To send data, write the data to be sent to the lower byte of I²C _DATA_CMD, and the command control bit CMD (I²C _DATA_CMD[8]) must be set to 0. Similarly, for a read operation, write ignorable data to the lower byte of I²C _DATA_CMD, and the command control bit CMD (I²C _DATA_CMD[8]) must be set to 1. I²C continuously initiates transmission only when there is a command in the send FIFO. If the send FIFO is empty and the STOP bit (I²C _DATA_CMD[9]) is 1, I²C sends a STOP condition to end the transmission; if the STOP bit (I²C _DATA_CMD[9]) is 0, SCL remains low until the send FIFO is written with data again.

3) Abandon transmission

The ABORT control bit in the I²C _ENABLE register allows the transmission to be aborted before all the data

in the TX FIFO is transmitted. In response to the abort request, the controller generates a STOP condition on the I²C bus, and then the TX FIFO is refreshed. Aborting the transmission is only allowed in the master mode.

4) Management of Tx FIFO and generation of start, stop, and restart signals

When the Tx FIFO becomes empty, the host generates a stop signal. If the RESTART generation is enabled, the host generates a RESTART signal when the Tx FIFO switches from read to write or from write to read. If it is not enabled, the host first generates a stop signal and then a start signal when the above situations occur. Note: In the design, if IC_EMPTYFIFO_HOLD_MASTER_EN = 0 is adopted, there will be no RESTART signal generated. Only when the FIFO is empty will a STOP signal be generated, and when there is data in the FIFO, a START signal will be generated.

REGISTER LIST

Register Mapping

I²C _BASE : 0x4000A000

I²C register:

Offset	Register Name	Description	Reset Value
0x00	I²C _CON	I²C control register	0x0000_007d
0x04	I²C _TAR	I²C Target Address Register	0x0000_0055
0x08	I²C _SAR	I²C slave address register	0x0000_0055
0x10	I²C _DATA_CMD	I²C receive and transmit data buffer and command	0x0000_0000
0x14	I²C _SS_SCL_HCNT	The high-level count value of the I²C clock SCL in standard mode	0x0000_007d
0x18	I²C _SS_SCL_LCNT	The low-level count value of the I²C clock SCL in standard mode	0x0000_0093
0x1c	I²C _FS_SCL_HCNT	The high-level count value of the I²C clock SCL in the fast mode	0x0000_0013
0x20	I²C _FS_SCL_LCNT	The low-level count value of the I²C clock SCL in fast mode	0x0000_0029
0x2c	I²C _INT_STAT	I²C Interrupt Status Register	0x0000_0000
0x30	I²C _INT_MASK	I²C interrupt mask register	0x0000_48ff
0x34	I²C _RAW_INT_ST	I²C Raw Interrupt Status Bits	0x0000_0000
0x38	I²C _RX_TL	I2C Receive FIFO Threshold Register	0x0000_0000
0x3c	I²C _TX_TL	I²C Transmit FIFO Threshold Register	0x0000_0000
0x40	I²C _CLR_INT	Clear the combined interrupt and independent interrupt registers.	0x0000_0000
0x44	I²C _CLR_RX_UD	Clear the RX_UNDER interrupt	0x0000_0000
0x48	I²C _CLR_RX_OV	Clear the RX_OVER interrupt	0x0000_0000
0x4c	I²C _CLR_TX_OV	Clear the TX_OVER interrupt	0x0000_0000
0x50	I²C _CLR_RD_REQ	Clear the RD_REQ interrupt	0x0000_0000
0x54	I²C _CLR_TX_ABRT	Clear the TX_ABRT interrupt	0x0000_0000

0x58	I²C _CLR_RX_DONE	Clear the RX_DONE interrupt	0x0000_0000
0x5c	I²C _CLR_ACTIVITY	Clear the ACTIVITY interrupt	0x0000_0000
0x60	I²C _CLR_STOP_DET	Clear the STOP_DET interrupt	0x0000_0000
0x64	I²C _CLR_START_DET	Clear the START_DET interrupt	0x0000_0000
0x68	I²C _CLR_GEN_CALL	Clear the GEN_CALL interrupt	0x0000_0000
0x6c	I²C _ENABLE	I²C Enable Register	0x0000_0000
0x70	I²C _STATUS	I²C status register	0x0000_0006
0x74	I²C _TXFLR	Send FIFO level register	0x0000_0000
0x78	I²C _RXFLR	Receive FIFO level register	0x0000_0000
0x7c	I²C _SDA_HOLD	SDA Hold Time Control Register	0x0000_0005
0x80	I²C_TX_ABRT_SOURCE	I²C transmission discards the source register	0x0000_0000
0x94	I²C _SDA_SETUP	I²C SDA Setup Time Register	0x0000_0028
0x98	I²C _ACK_GEN_CALL	I²C broadcast call response register	0x0000_0001
0x9C	I²C _ENABLE_STATUS	I²C Enable Status Register	0x0000_0000
0xf4	I²C _COMP_PARAM_1	I²C parameter register	0x0007_07AA

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write, WC: Write clear

I²C _CON: Address (0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	N/A	Not used	0x0
7	STOP_DET_IFADDRESSED	R/W	Generation of stop_det interrupt; (slave mode) 1: Address matching generates stop_det interrupt; 0: No address matching is required to generate stop_det interrupt. Note: After detecting the stop signal	0x0
6	IC_SLAVE_DISABLE	R/W	Slave enable 0: Slave enable is on; 1: Slave enable is off.	0x1
5	IC_RESTART_EN	R/W	Whether to send the restart signal when acting as the host: 0: Disable the restart function 1: Enable the restart function	0x1
4	IC_10BITADDR_MASTER	R/W	Host address bit width selection: 0: 7-bit address 1: 10-bit address	0x1
3	IC_10BITADDR_SLAVE	R/W	Slave address bit width selection; 0: 7-bit address.	0x1

			1: 10-bit address.	
2:1	SPEED	R/W	Rate selection 0x1: Standard mode (0 to 100 Kb/s) 0x2: Fast mode (≤ 400 Kb/s) Others: Disabled	0x2
0	MASTER_MODE	R/W	Host enable 0: Host enable is off; 1: Host enable is on.	0x1

I²C _TAR: Address (0x04)

Bit	Symbol	R/W	Description	Default
31:12	Reserved	R/W	Not used	0x0
11	SPECIAL	R/W	Determine whether to use General Call, START byte 0: Ignore bit10 and use this register normally 1: Use Device_ID or bit10	0x0
10	GC_OR_START	R/W	If bit11 = 1 and bit13 = 0, then this bit determines whether to use the START byte or General Call. 0: General Call 1: START byte	0x0
9:0	IC_TAR	R/W	Store the target address	0x55

I²C _SAR: Address (0x08)

Bit	Symbol	R/W	Description	Default
31:10	Reserved	R/W	Not Used	0x0
9:0	IC_SAR	R/W	Save the device address of DW_apb_I²C when it is in slave mode. If a 7-bit address is used, only seven bits are needed.	0x55

I²C _DATA_CMD : Address (0x10)

Bit	Symbol	R/W	Description	Default
31:10	Reserved	R/W	Not used	0x0
9	STOP	W	IC_EMPTYFIFO_HOLD_MASTER_EN = 1 (Stop signal sent after data transmission and reception) 0: No stop signal is sent regardless of whether the Tx FIFO is empty or not; instead, SCL is pulled low to wait for a new command. 1: A stop signal is sent regardless of whether the Tx FIFO is empty or not; if	0x0

			sent when not empty, the master will immediately attempt to start a new transmission.	
8	CMD	W	Host mode read/write control bit: 0: Write 1: Read	0x0
7:0	DAT	R/W	Data transmitted on the IIC bus	0x0

I²C _SS_SCL_HCNT: Address (0x14)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	R/W	Not used	0x0
15:0	IC_SS_SCL_HCNT	R/W	In standard mode, the high-level count of SCL Note: The minimum value is 6	0x28

I²C _SS_SCL_LCNT : Address (0x18)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	R/W	Not used	0x0
15:0	IC_SS_SCL_LCNT	R/W	In standard mode, the low-level count of SCL Note: The minimum value is 8	0x2f

I²C _FS_SCL_HCNT: Address (0x1c)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	R/W	Not used	0x0
15:0	IC_FS_SCL_HCNT	R/W	In the fast mode (400k), the high-level count of SCL, note: the minimum is 6.	0x6

I²C _FS_SCL_LCNT : Address (0x20)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	R/W	Not used	0x0
15:0	IC_FS_SCL_LCNT	R/W	In the fast mode (400k), the low-level count of SCL, note: the minimum is 8.	0xd

I²C _INT_STAT : Address (0x2c)

Bit	Symbol	R/W	Description	Default
31:12	Reserved	R	Not used	0x0
11	R_GEN_CALL	R	Whether an interrupt is generated for RAW_INT_STAT[11];	x

			1: This interrupt is generated; 0: This interrupt is not generated;	
10	R_START_DET	R	Whether the interrupt of RAW_INT_STAT[10] is generated; 1: This interrupt is generated; 0: This interrupt is not generated;	x
9	R_STOP_DET	R	Whether an interrupt is generated for RAW_INT_STAT[9]; 1: This interrupt is generated; 0: This interrupt is not generated;	x
8	R_ACTIVITY	R	Whether an interrupt is generated for RAW_INT_STAT[8]; 1: This interrupt is generated; 0: This interrupt is not generated;	x
7	R_RX_DONE	R	Whether an interrupt is generated for RAW_INT_STAT[7]; 1: This interrupt is generated; 0: This interrupt is not generated;	x
6	R_TX_ABRT	R	Whether an interrupt is generated for RAW_INT_STAT[6]; 1: This interrupt is generated; 0: This interrupt is not generated;	x
5	R_RD_REQ	R	Whether an interrupt is generated for RAW_INT_STAT[5]; 1: This interrupt is generated; 0: This interrupt is not generated;	x
4	R_TX_EMPTY	R	Whether an interrupt is generated for RAW_INT_STAT[4]; 1: This interrupt is generated; 0: This interrupt is not generated;	x
3	R_TX_OVER	R	Whether an interrupt is generated for RAW_INT_STAT[3]; 1: This interrupt is generated; 0: This interrupt is not generated;	x
2	R_RX_FULL	R	Whether an interrupt is generated for RAW_INT_STAT[2]; 1: This interrupt is generated; 0: This interrupt is not generated;	x
1	R_RX_OVER	R	Whether an interrupt is generated for RAW_INT_STAT[1]; 1: This interrupt is generated; 0: This interrupt is not generated;	x

0	R_RX_UNDER	R	Whether an interrupt is generated for RAW_INT_STAT[0]; 1: This interrupt is generated; 0: This interrupt is not generated;	x

I²C _INT_MASK : Address (0x30)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:12	Reserved	R/W	Not used	0x0
11	M_GEN_CALL	R/W	GEN_ALL interrupt enable; 1: Enable GEN_ALL interrupt; 0: Disable GEN_ALL interrupt;	0x1
10	M_START_DET	R/W	Enabling or disabling the start or restart interrupt; 1: Enable the start/restart interrupt; 0: Disable the start/restart interrupt	0x1
9	M_STOP_DET	R/W	Stop (Master/Slave) interrupt enable 1: Enable stop interrupt 0: Disable stop interrupt	0x1
8	M_ACTIVITY	R/W	DW_apb_ I²C interrupt enable status; 1: Interrupt enable is on; 0: Interrupt enable is off	0x1
7	M_RX_DONE	R/W	The interrupt enable of RAW_INT_STAT[7]; 1: Enable the interrupt; 0: Disable the interrupt	0x1
6	M_TX_ABRT	R/W	The interrupt enable of RAW_INT_STAT[6]; 1: Enable the interrupt; 0: Disable the interrupt	0x1
5	M_RD_REQ	R/W	The interrupt enable of RAW_INT_STAT[5]; 1: Enable the interrupt; 0: Disable the interrupt	0x1
4	M_TX_EMPTY	R/W	The interrupt enable of RAW_INT_STAT[4]; 1: Enable the interrupt; 0: Disable the interrupt	0x1
3	M_TX_OVER	R/W	The interrupt enable of RAW_INT_STAT[3]; 1: Enable the interrupt; 0: Disable the interrupt	0x1

2	M_RX_FULL	R/W	The interrupt enable of RAW_INT_STAT[2]; 1: Enable the interrupt; 0: Disable the interrupt	0x1
1	M_RX_OVER	R/W	The interrupt enable of RAW_INT_STAT[1]; 1: Enable the interrupt; 0: Disable the interrupt	0x1
0	M_RX_UNDER	R/W	The interrupt enable of RAW_INT_STAT[0]; 1: Enable the interrupt; 0: Disable the interrupt	0x1

I²C _RAW_INT_STAT: Address (0x34)

Bit	Symbol	R/W	Description	Default
31:12	Reserved	R	Not used	0x0
11	GEN_CALL	R	Check if there is GEN_ALL. Notes: DW_apb_ I²C enable must be turned off or the CPU reads the cleared 0 bit to clear it.	0x0
10	START_DET	R	Detect whether there is a start or restart (either the master or the slave).	0x0
9	STOP_DET	R	Detect whether there is a stop (either on the host or the slave).	0x0
8	ACTIVITY	R	There are four ways to reset DW_apb_ I²C if it is active: 1. Disable the DW_apb_ I²C enable. 2. Read IC_CLR_ACTIVITY. 3. Read IC_CLR_INTR. 4. Perform a system reset.	0x0
7	RX_DONE	R	When acting as the slave sender, if the master does not send an ACK signal after one byte is sent, this signal will be set to 1, indicating that the transmission is complete.	0x0
6	TX_ABRT	R	The transmission abort flag will clear the Tx FIFO. The IC_TX_ABRT_SOURCE register will indicate the reason for the transmission interruption.	0x0
5	RD_REQ	R	When DW_apb_ I²C acts as a slave, this bit will be set to 1 when other masters want to obtain data from this device. The processor needs to handle this interrupt and write the data into	0x0

			DATA_CMD. The interrupt will be cleared only when the processor reads the CLR register.	
4	TX_EMPTY	R	This interrupt occurs when the sending cache reaches or drops below the threshold of Tx. (When it is above the threshold, it will be automatically cleared.)	0x0
3	TX_OVER	R	An interrupt is generated when the Tx FIFO is already full and the processor still attempts to write data to the DATA_CMD register. (The interrupt is cleared when the state machine returns to IDLE and ic_en = 0.)	0x0
2	RX_FULL	R	When the receive buffer reaches or exceeds the threshold of Rx, this interrupt will be generated. (It will be automatically cleared when it is below this value.)	0x0
1	RX_OVER	R	This interrupt occurs when the Rx FIFO is already full and additional data is received. (The interrupt is cleared when the state machine returns to IDLE and ic_en = 0) Note: The additional received data will be lost.	0x0
0	RX_UNDER	R	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. (The state machine returns to IDLE and ic_en = 0 will clear the interrupt.)	0x0

I²C _RX_TL : Address (0x38)

Bit	Symbol	R/W	Description	Default
31:8	Reserved	N/A	Not used	0x0
7:0	RX_TL	R/W	Set the threshold for triggering the Rx_FULL interrupt Notes: The set value will be incremented by 1. The value is not allowed to exceed the depth of the FIFO.	0x0

I²C _TX_TL: Address (0x3c)

Bit	Symbol	R/W	Description	Default

31:8	Reserved	N/A	Not used	0x0
7:0	TX_TL	R/W	Set the threshold for triggering the TX_EMPTY interrupt	0x0

I²C _CLR_INTR: Address (0x40)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	N/A	Not used	0x0
0	CLR_INTR	R	Clear the interrupt in the IC_TX_ABRT_SOURCE register; Reading this bit can clear the interrupt.	0x0

I²C _CLR_RX_UNDER: Address (0x44)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	N/A	Not used	0x0
0	CLR_RX_UNDER	R	Read the register to clear the RX_UNDER interrupt; in the register IC_RAW_INTR_STAT.	0x0

I²C _CLR_RX_OVER: Address (0x48)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	N/A	Not used	0x0
0	CLR_RX_OVER	R	Read the register to clear the RX_OVER interrupt; in the register IC_RAW_INTR_STAT.	0x0

I²C _CLR_TX_OVER: Address (0x4c)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	N/A	Not used	0x0
0	CLR_TX_OVER	R	Read the register to clear the TX_OVER interrupt; in the IC_RAW_INTR_STAT register	0x0

I²C _CLR_RD_REQ : Address (0x50)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	R	Not Used	0x0
0	CLR_RD_REQ	R	Read the register to clear the RD_REQ interrupt; in the register IC_RAW_INTR_STAT.	0x0

I²C _CLR_TX_ABRT : Address (0x54)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	R	Not Used	0x0
0	CLR_TX_ABRT	R	Read the register to clear the TX_ABRT interrupt; check the IC_RAW_INTR_STAT and IC_TX_ABRT_SOURCE registers.	0x0

I²C _CLR_RX_DONE : Address (0x58)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	R	Not Used	0x0
0	CLR_RX_DONE	R	Read the register to clear the RX_DONE interrupt; in the register IC_RAW_INTR_STAT.	0x0

I²C _CLR_ACTIVITY: Address (0x5c)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	R	Not Used	0x0
0	CLR_ACTIVITY	R	Read the register to clear the ACTIVITY interrupt; in the register IC_RAW_INTR_STAT.	0x0

I²C _CLR_STOP_DET: Address (0x60)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	R	Not Used	0x0
0	CLR_STOP_DET	R	Read the register to clear the STOP_DET interrupt; in the register IC_RAW_INTR_STAT.	0x0

I²C _CLR_START_DET: Address (0x64)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	R	Not Used	0x0
0	CLR_START_DET	R	Read the register to clear the START_DET interrupt; in the register IC_RAW_INTR_STAT.	0x0

I²C _CLR_GEN_CALL: Address (0x68)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	R	Not Used	0x0
0	CLR_GEN_CALL	R	Read the register to clear the GEN_CALL interrupt; in the register	0x0

			IC_RAW_INTR_STAT.	

I²C _ENABLE: Address (0x6c)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:2	Reserved	N/A	Not used	0x0
1	ABORT	R/W	Send Termination Enable: 0: Do not perform termination transmission 1: Terminate transmission in progress Note: After initiating termination, the controller will send a stop signal and clear the Tx FIFO, and simultaneously trigger an interrupt; the software cannot clear this bit, and it will automatically reset to zero after the termination is completed. Transmission can only be stopped after the ENABLE is enabled.	0x0
0	ENABLE	R/W	Control whether DW_apb_ I²C is enabled: 0: Disable (Both the Tx and Rx FIFOs are in erased state) 1: Enable Notes: If the enable is turned off, the TX FIFO and RX FIFO are erased; IC_INTR_STAT remains active until the IIC transitions to the IDLE state.	0x0

I²C _STATUS : Address (0x70)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:7	Reserved	N/A	Not used	0x0
6	SLV_ACTIVITY	R	The working status of the slave machine: 0: Idle 1: Working.	0x0
5	MST_ACTIVITY	R	Host working status: 0: Idle 1: Working.	0x0
4	RFF	R	The flag indicating that the Rx FIFO is completely full: 0: The Rx FIFO is not full. 1: The Rx FIFO is full. Notes: When there is still some free space in the FIFO, this flag bit will be	0x0

			automatically cleared.	
3	RFNE	R	No empty flag for Rx FIFO: 0: Rx FIFO is empty 1: Rx FIFO is not empty Note: This flag will be automatically cleared when the Rx FIFO becomes empty.	0x0
2	TFE	R	Flag indicating that the Tx FIFO is completely empty: 0: The Tx FIFO is not empty 1: The Tx FIFO is empty Note: This flag is automatically cleared when the Tx FIFO becomes non-empty.	0x1
1	TFNF	R	The flag indicating that the Tx FIFO is not full: 0: The Tx FIFO is full. 1: The Tx FIFO is not full. Note: This flag will be automatically cleared when the Tx FIFO is full.	0x1
0	ACTIVITY	R	I²C status	0x0

I²C _TXFLR: Address (0x74)

Bit	Symbol	R/W	Description	Default
31:4	Reserved	N/A	Not used	0x0
3::0	TXFLR	R	How many valid data are there in the Tx FIFO? Note: There are three ways to clear the register: 1. DW_apb_ I²C enable is turned off. 2. Transmission is terminated (TX_ABRT). 3. Slave block transmission is terminated.	0x0

I²C _RXFLR: Address (0x78)

Bit	Symbol	R/W	Description	Default
31:4	Reserved	N/A	Not used	0x0
3:0	RXFLR	R	How many valid data are there in the Rx FIFO? Note: There are three ways to clear the register:	0x0

			1. DW_apb_ I²C enable is turned off. 2. Transmission is terminated (TX_ABRT).	

I²C _SDA_HOLD: Address (0x7c)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:24	Reserved	N/A	Not Used	0x0
23:16	IC_SDA_RX_HOLD	R/W	The hold time of SDA at the time of reception	0x0
15:0	IC_SDA_TX_HOLD	R/W	The hold time of SDA when sending.	0x1

I²C _TX_ABRT_SOURCE: Address (0x80)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:27	Reserved	N/A	Not used	0x0
26:23	TX_FLUSH_CNT	R	Indicates the amount of data in the Tx FIFO that was erased due to transmission interruption. Note: Disabling the I²C function can clear it.	0x0
22:17	Reserved	N/A	Not used	0x0
16	ABRT_USER_ABRT	R	The host detects the transmission termination (IC_ENABLE[1]) Notes: Only applicable to the host sender	0x0
15	ABRT_SLVRD_INTX	R	When the processor was handling the data transmission from the host to other hosts, the user wrote a 1 into the eighth bit of IC_DATA_CMD. Note: Can only act as a slave sender.	0x0
14	ABRT_SLV_ARBLOST	R	When the slave device loses the bus during the process of transmitting data to other master devices, bit 12 will also be set to 1. Note: It can only act as a slave sender.	0x0
13	ABRT_SLVFLUSH_TXFIFO	R	When the slave device receives a read command but there is still data in the Tx FIFO, it issues a Tx_ABRT interrupt to clear the old data. Note: It can only be used as a slave sender.	0x0

12	ARB_LOST	R	If the master arbitration fails or bit 14 is set to 1, the slave sends a lost bus. Note: As the sender	0x0
11:8	Reserved	N/A	Not used	0x0
7	ABRT_SBYTE_ACKDET	R	Abnormal flag. The host sent "start" but received no response.	0x0
6:5	Reserved	N/A	Not used	0x0
4	ABRT_GCALL_NOACK	R	Abnormal flag. The host sent a broadcast communication but received no response from any slave on the bus.	0x0
3	ABRT_TXDATA_NOACK	R	Abnormal flag. The host received the ACK from the slave for the address but did not receive the ACK for the data.	0x0
2	ABRT_10ADDR2_NOACK	R	Abnormal flag. The host sent the second frame address in the 10-bit address mode but received no response.	0x0
1	ABRT_10ADDR1_NOACK	R	Abnormal flag. The host sent the first frame address in the 10-bit address mode but received no response.	0x0
0	ABRT_7B_ADDR_NOACK	R	Exception flag. The host did not receive a response when sending the address in 7-bit address mode.	0x0

I²C _SDA_SETUP : Address (0x94)

Bit	Symbol	R/W	Description	Default
31:8	Reserved	N/A	Not used	0x0
7:0	SDA_SETUP	R/W	After the data changes, set the low-level delay time of SCL. Notes: The minimum value is 2. Calculation method: [(IC_SDA_SETUP - 1) * (ic_clk_period)]	0x64

I²C_ACK_GEN_CALL (0x98)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	N/A	Not used	0x0

0	ACK_GEN_CALL	R/W	ACK response to general call; 1: Received the broadcast call and sent an ACK response; 0: Did not receive the broadcast call;	0x1

I²C_ENABLE_STATUS: Address (0x9c)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:3	Reserved	N/A	Not used	0x0
2	SLV_RX_DATA_LOST	R	Data loss status of the slave machine received;	0x0
1	SLV_DISABLED_WHILE_BUSY	R	Slave failure status during transmission or reception process;	0x0
0	IC_EN	R	The status of IC_EN	0x0

IC_COMP_PARAM_1: Address (0xf4)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:24	Reserved	N/A	Not used	0x0
23:16	TX_BUFFER_DEPTH	R	The depth of the Tx buffer (2 to 256)	0x7
15:8	RX_BUFFER_DEPTH	R	The depth of the Rx buffer (2 to 256)	0x7
7	ADD_ENCODED_PARAMS	R	Can the parameters be encoded? 0: False 1: True	0x1
6	HAS_DMA	R	Is DMA enabled? 0: False 1: True	0x0
5	INTR_IO	R	Whether to use independent interrupts or combined interrupts for interruption 0: Independent 1: Combined	0x1
4	HC_COUNT_VALUES	R	Whether to enable the change amount 0: False 1: True	0x0
3:2	MAX_SPEED_MODE	R	Communication rate: 0x0 = reserve 0x1 = Standard 0x2 = Fast 0x3 = High Speed	0x2

1:0	APB_DATA_WIDTH	R	APB data bus width: 0x0 = 8bites 0x1 = 16bites 0x2 = 32bites 0x3 = reserve	0x2

SPIM

OVERVIEW

This module is a SPI host interface module, used for communication with external SPI interface devices. The default mode is Motorola SPI mode. The SPI interface supports full-duplex and half-duplex communication, and also supports multi-master transmission operations.

FEATURES

- 32-bit APB bus interface
- 8-depth transmit FIFO and receive FIFO
- Configurable data frame format
- Configurable polarity and phase of serial clock
- Supports multi-master conflict interrupt, FIFO overflow interrupt, and transmission error interrupt
- Only supports continuous transmission mode

FUNCTIONAL DESCRIPTION

SPI is a high-speed, synchronous, full duplex communication bus that can communicae between MCU and external devices.

Transmission Mode

The SPI module supports four transmission modes, transmit and receive, transmit only, and receive only. Different transmission modes can be set by TMOD of register CTRLR0.

When TMOD is configured to 2'b00, both the sending and receiving circuits will work normally, and data transmission will be carried out according to the set frame format. The transmitted data will be sent out from the transmit FIFO and sent to the slave through the MOSI data line, and the slave will reply to the data through the MISO data line. After receiving the data, it will be stored in the receive FIFO.

When TMOD is configured as 2'b01, the received data is invalid and should not be stored in the receiving IFFO. The transmission of data should proceed normally, and interrupts related to reception should be blocked in this mode.

When TMOD is configured as 2'b10, the data transmission is invalid, and any related interrupts should be blocked. Receive data normally.

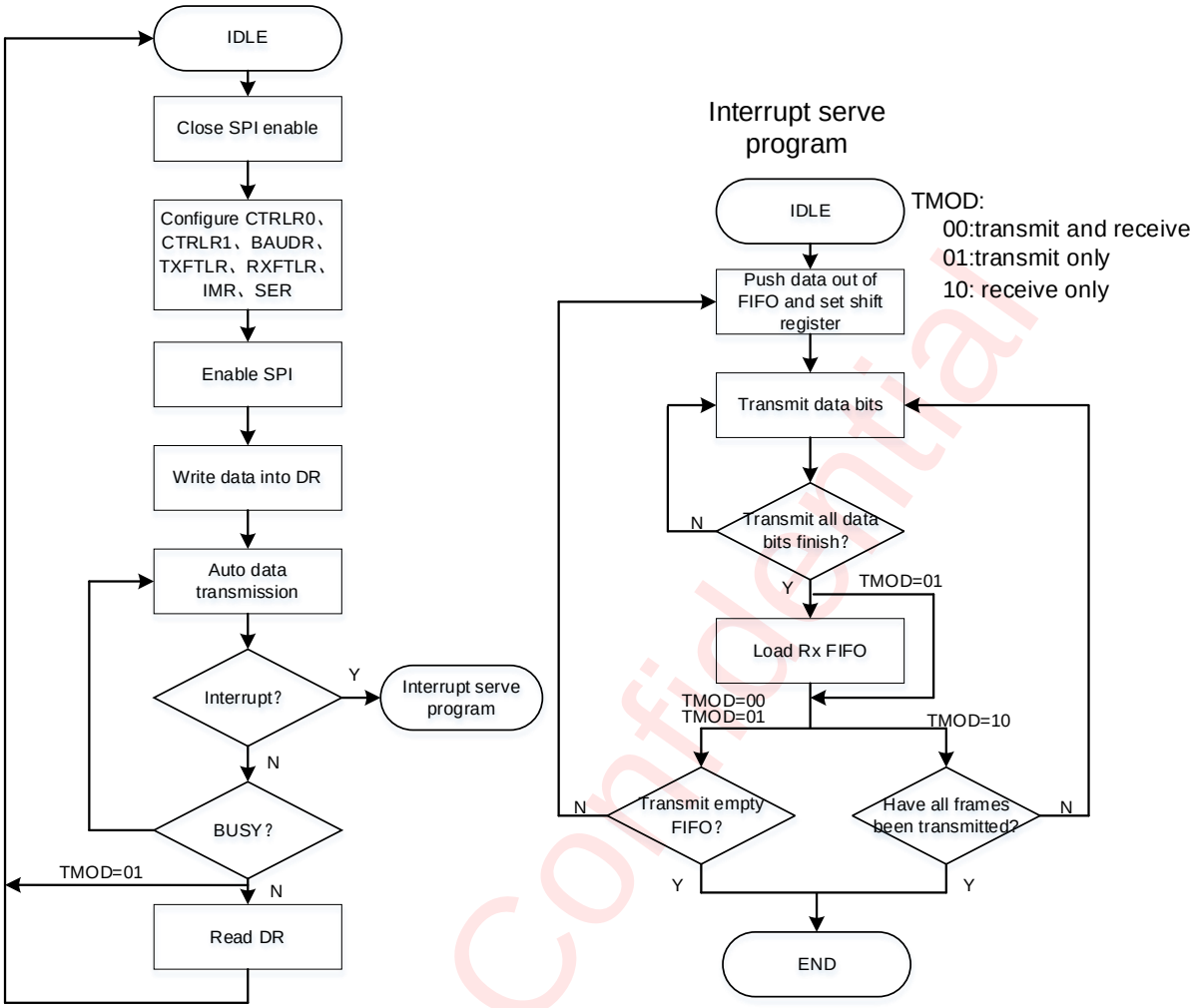
Interrupt

The SPI module supports multiple maskable interrupts, and each interrupt combination generates an interrupt signal for the SPI module, which is sent to the interrupt controller. The specific interrupt description is shown in the table below.

Interrupt	Description
Transmit FIFO null interrupt	When the data in the FIFO is less than or equal to the set threshold, this interrupt will be generated. When data is written into the transmit

	FIFO and the number of data exceeds the set threshold, this interrupt will be automatically cleared.
Transmit FIFO overflow interrupt	When the sending FIFO is full and the user writes data to the FIFO again, this interrupt will be triggered and the write operation will be invalid. Clearing this interrupt requires reading the register TXOICR.
Receive FIFO full interrupt	When the received data in the FIFO is greater than or equal to the set threshold plus 1, this interrupt will be generated. When the received FIFO data is read out and the number of data is less than the set threshold, this interrupt will be automatically cleared.
Receive FIFO overflow interrupt	When the receive FIFO is in a full state and the user writes data to the FIFO again, this interrupt will be triggered and the write operation will be invalid. Clearing this interrupt requires reading the register RXOICR.
Receive FIFO underflow interrupt	When the receive FIFO is in an empty state, a user reading the receive FIFO will generate this interrupt. Clearing this interrupt requires reading the register RXUICR.
Multi host conflict interrupt	When another host on the bus selects this module as a slave and transfers data, this interrupt will be triggered. Clearing this interrupt requires reading the register MSTICR.
Combination interrupt	The combination interrupt of all the above interrupts, blocking this interrupt requires blocking all the above interrupts.

SPI configuration



REGISTER LIST

Register Mapping

SPI_BASE : 0x40004900

SPI register:

Offset	Register Name	R/W	Description	Reset Value
0x00	CTRLR0	RW	Control Register 0	0x070000
0x04	CTRLR1	RW	Control Register 1	0x00
0x08	SSIENR	RW	SPI Enable Register	0x00
0x0C	MWCR	RW	Microwire control register	0x00
0x10	SER	RW	Slave Enable Register	0x00
0x14	BAUDR	RW	Baud Rate Configuration Register	0x00
0x18	TXFTLR	RW	Send the FIFO threshold register	0x00
0x1C	RXFTLR	RW	Receive FIFO Threshold Register	0x00
0x20	TXFLR	RO	Register for the number of valid data in the sending FIFO	0x00

0x24	RXFLR	RO	Register for the number of valid data in the receiving FIFO	0x00
0x28	SR	RW	Status register	0x06
0x2C	IMR	RW	Interrupt Mask Register	0x3F
0x30	ISR	RO	Interrupt Status Register	0x00
0x34	RISR	RO	Original Interrupt Status Register	0x00
0x38	TXOICR	RW	Clear the FIFO overflow interrupt register.	0x00
0x3C	RXOICR	RW	Clear the receive FIFO overflow interrupt register.	0x00
0x40	RXUICR	RW	Clear the receive FIFO underflow interrupt register.	0x00
0x44	MSTICR	RW	Clear multiple host interrupt registers	0x00
0x48	ICR	RW	Interrupt Clear Register	0x00
0x60~0xEC	DR	RW	Data register	0x00
0xf0	RX_SAMPLE_DLY	RW	Reception sampling delay control period register	0x00

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write

CTRLR0: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:23	Reserved	RW	Not used	0
22:21	SPI_FRF	RW	Not Use	0x0
20:16	DFS_32	RW	Used to configure the size of data transmission in 32-bit mode: 0-2: Not used; 3: 4-bit transmission; 4: 5-bit transmission; … 31:32-bit transmission	0x7
15:12	CFS	RW	Frame size control, select the length of the Microwire protocol control word. 0: 1-bit control word; 1: 2-bit control word; ... n: n + 1-bit control word;	0x0
11	SRL	RW	Shift register control, only used in test mode. When this is set to 1, the output of the shift register at the transmitting end will be connected to the input of the receiving shift register. 0: Normal mode;	0x0

			1: Test mode;	
10	SLV_OE	RW	Not Use in Master	0x0
9:8	TMOD	RW	Transmission mode configuration: 00: Send and receive 01: Send only 10: Receive only 11: EEPROM read	0x0
7	SCPOL	RW	The polarity of the serial clock level is only valid when FRF is configured as Motorola SPI. 0: The SPI clk is low in the idle state. 1: The SPI clk is high in the idle state.	0x0
6	SCPH	RW	The serial clock phase is only valid when FRF is configured as Motorola SPI. 0: Data is sampled on the first clock edge. 1: Data is sampled on the second clock edge.	0x0
5:4	FRF	RW	Frame format selection, select SPI operation mode 00: Motorola mode 01: TI mode 10: Microwire mode 11: Not used	0x0
3:0	DFS	RW	The size of the data frame, the reserved data configured (0~2) will not take effect. 0~2: Not used 3: 4 bits … n:n+1 bit	0x0

CTRLR1: Address(0x04)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RW	Not used	0
15:0	NDF	RW	The number of data frames: when TMOD is 10 or 11, that is, when SPI is in only-receiving or reading EEPROM mode, this register is used to set the number of consecutive data frames to be received. When the number of received data frames equals the value of this register plus 1, SPI stops receiving data.	0x0

SSIENR: Address(0x08)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	RW	Not used	0

0	SSI_EN	RW	SPI Enable Control 0: Disable. All transmissions will stop, and the data in the send and receive FIFOs will be cleared. (When disabled, the SPI Sleep signal will be set, and the system can turn off the SPI module's clock.) 1: Enable.	0x0

MWCR: Address(0x0C)

Bit	Symbol	R/W	Description	Default
31:3	Reserved	RW	Not used	0
2	MHS	RW	Microwire handshake signal control. When this bit is set, the SPI module will check the ready state of the slave. After the transmission is completed, the BUSY signal in the SR register will be cleared. 0: Disable Microwire handshake signal 1: Enable Microwire handshake signal	0x0
1	MDD	RW	The Microwire control signal configures the direction of data transmission. 0: Receive 1: Transmit	0x0
0	MWMOD	RW	Microwire transmission mode control, configure the Microwire transmission as sequential transmission or non-sequential transmission. 0: Non-continuous transmission, each transmission data must have a control word. 1: Continuous transmission, one control word is needed for transmitting a block of data.	0x0

SER: Address(0x10)

Bit	Symbol	R/W	Description	Default
31:3	Reserved	RW	Not used	0
2:0	SER	RW	Slave enable control bits: 000: Disable all slaves 001: Enable CS0 slave 010: Enable CS1 slave 100: Enable CS2 slave	0x0

BAUDR: Address(0x14)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RW	Not used	0
15:0	SCKDV	RW	The SPI module clock divider, the 0th bit cannot be written, so the value of SCLKDV is an even number within the range of 2 to 65534. Fsclk_clk = Fclk	0x0

			/SCKDV	

TXFTLR: Address(0x18)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:3	Reserved	RW	Not used	0
2:0	TFT	RW	Send FIFO threshold: When the data in the send FIFO is less than or equal to the set value, an interrupt is triggered.	0x0

RXFTLR: Address(0x1C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:3	Reserved	RW	Not used	0
2:0	RFT	RW	The receive FIFO threshold triggers an interrupt when the data in the transmit FIFO is greater than or equal to the set value.	0x0

TXFLR: Address(0x20)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:4	Reserved	RW	Not used	0
3:0	TXTFL	RW	The number of valid data in the sending FIFO	0x0

RXFLR: Address(0x24)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:4	Reserved	RW	Not used	0
3:0	RXTFL	RW	The number of valid data in the receiving FIFO	0x0

SR: Address(0x28)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:7	Reserved	RO	Not used	0
6	DCOL	RO	The data conflict error flag bit will be set if another host pulls up the ss_in_n signal. If the transmission process is terminated, this bit is used to notify the CPU that the transmission was not completed. Reading this bit will clear the flag.	0x0
5	TXE	RO	Not Use in Master device	0x0
4	REF	RO	The receive FIFO full flag bit is set to 1 when the receive FIFO is full, and it is automatically cleared to 0 when the receive FIFO is not full. 0: FIFO is not full; 1: FIFO is full;	0x0

3	RFNE	RO	Receive FIFO not empty flag bit. When the receive FIFO is in a non-empty state, this bit is set to 1. 0: FIFO is empty 1: FIFO is non-empty	0x0
2	TFE	RO	Transmit FIFO not empty flag bit. When the transmit FIFO is in a non-empty state, this bit is set to 1. 0: FIFO is non-empty 1: FIFO is empty	0x1
1	TFNF	RO	Transmit FIFO not full flag bit. When the transmit FIFO is not full, this bit is set to 1. 0: FIFO is full 1: FIFO is not full	0x1
0	BUSY	RO	The busy flag bit is set to 1 when SPI is transmitting data. 0: SPI is idle or disabled. 1: SPI is in the process of data transmission.	0x0

IMR: Address(0x2C)

Bit	Symbol	R/W	Description	Default
31:6	Reserved	RW	Not used	0
5	MSTIM	RW	Multi-host conflict interrupt mask bit 0: Masked 1: Unmasked	0x1
4	RXFIM	RW	Receive FIFO full interrupt mask bit 0: Masked 1: Unmasked	0x1
3	RXOIM	RW	Receive FIFO overflow interrupt mask bit 0: Masked 1: Unmasked	0x1
2	RXUIM	RW	Receive FIFO underflow interrupt mask bit 0: Masked 1: Unmasked	0x1
1	TXOIM	RW	Set the FIFO overflow interrupt mask bit for transmission 0: Masked 1: Unmasked	0x1
0	TXEIM	RW	Set the FIFO empty interrupt mask bit for transmission 0: Masked 1: Unmasked	0x1

ISR: Address(0x30)

Bit	Symbol	R/W	Description	Default

31:6	Reserved	RO	Not used	0
5	MSTIS	RO	Multi-host conflict interrupt flag bit	0x0
4	RXFIS	RO	Receive FIFO full interrupt flag bit	0x0
3	RXOIS	RO	The overflow interrupt flag bit of the receiving FIFO	0x0
2	RXUIS	RO	Receive FIFO underflow interrupt flag bit	0x0
1	TXOIS	RO	Transmit FIFO overflow interrupt flag bit	0x0
0	TXEIS	RO	The interrupt flag bit for the empty transmission FIFO	0x0

RISR: Address(0x34)

Bit	Symbol	R/W	Description	Default
31:6	Reserved	RO	Not used	0
5	MSTIR	RO	Original interrupt flag bit for multi-host conflict	0x0
4	RXFIR	RO	The original interrupt flag bit for the receive FIFO being full	0x0
3	RXOIR	RO	The original interrupt flag bit for the overflow of the receiving FIFO	0x0
2	RXUIR	RO	The original interrupt flag bit for the underflow of the receiving FIFO	0x0
1	TXOIR	RO	Send the original interrupt flag bit of FIFO overflow	0x0
0	TXEIR	RO	The original interrupt flag bit for an empty FIFO in transmission is sent.	0x0

TXOICR: Address(0x38)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	RO	Not used	0
0	TXOICR	RO	Clear the FIFO overflow interrupt by reading.	0x0

RXOICR: Address(0x3C)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	RO	Not used	0
0	RXOICR	RO	Enable the reception FIFO overflow interrupt and read to clear it.	0x0

RXUICR: Address(0x40)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	RO	Not used	0
0	RXUICR	RO	Clear the underflow interrupt of the receive FIFO and read it.	0x0

MSTICR: Address(0x44)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	RO	Not used	0
0	MSTICR	RO	Clear the multi-host conflict interrupt and read the clear.	0x0

ICR: Address(0x48)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:1	Reserved	RO	Not used	0
0	ICR	RO	Clear the interrupt register. Reading this register will clear the above four types of interrupts.	0x0

DR: Address(0x60~0xEC)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:0	DR	RW	Data register	0x0

RX_SAMPLE_DLY: Address(0xf0)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RW	Not used	0
7:0	RSD	RW	Receiving data sampling delay control	0x0

RCCM

OVERVIEW

The reset and Clock controller module (RCCM) is used to generate the reset and clock signals required for each module. In addition, the controller also provides reset source query and system reset functions.

FEATURES

- Providing clocks of TIM1, TIM2, TIM3, GPIOM, SPI, UART, I²C and WDT
- Providing reset of TIM1, TIM2, TIM3, GPIOM, SPI, UART, I²C and WDT
- Support reset source queries
- Support system reset

REGISTER LIST

Register Mapping

RCCM_BASE : 0x40004400

Offset	Register Name	Description	Reset Value
0x00	RCCM_CLKENR	Clock enable register	0
0x04	RCCM_RSTR	Reset register	0
0x08	RCCM_RSTCTRL	Reset control register	0x2

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write

clock enable register (RCCM_CLKENR): Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7	GPIOMEN	RW	GPIOM clock enabled 0: Disable 1: Enable	1
6	I2CEN	RW	I²C clock enabled 0: Disable 1: Enable	0
5	SPIEN	RW	SPI clock enabled 0: Disable 1: Enable	0
4	TIM1EN	RW	TIM1 clock enabled 0: Disable 1: Enable	0
3	TIM2EN	RW	TIM2 clock enabled 0: Disable 1: Enable	0
2	TIM3EN	RW	TIM3 clock enabled 0: Disable 1: Enable	0
1	UARTEN	RW	UART clock enabled 0: Disable 1: Enable	0
0	WDTEN	RW	WDT clock enabled 0: Disable 1: Enable	0

reset register (RCCM_RSTR): Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7	GPIOMRST	RW	GPIOM reset 0: Release 1: Reset GPIOM	
6	I2CRST	RW	I²C reset 0: Release 1: Reset I²C	
5	SPIRST	RW	SPI reset 0: Release	

			1: Reset SPI	
4	TIM1RST	RW	TIM1 reset 0: Release 1: Reset TIM1	0
3	TIM2RST	RW	TIM2 reset 0: Release 1: Reset TIM2	0
2	TIM3RST	RW	TIM3 reset 0: Release 1: Reset TIM3	0
1	UARTRST	RW	UART reset 0: Release 1: Reset UART	0
0	WDTRST	RW	WDT reset 0: Release 1: Reset WDT	0

reset ctrl register (RCCM_RSTCTRL): Address(0x08)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:4	Reserved	RO	Not used	0
3:1	RSTSRC	RW	System reset source 001: Default state, power on reset 010: WDT reset 011: CPU software reset 100: RCCM software reset	0x1
0	SFREQ	WO	System reset request 0: Disable 1: Enable	0

WDT

OVERVIEW

Watchdog Timer (WDT) can be used to detect and resolve faults caused by software errors; When the counter reaches the given timeout value, it triggers an interrupt or generates a system reset. The WDT is used to prevent system lock caused by device or program conflicts in SOC.

FEATURES

- 32 bits WDT count, operating in the system clock domain.
- The count decreases from the initial value to 0 when a timeout occurs.
- If a timeout occurs, it can be configured to reset the system directly by WDT or to generate an interrupt. If the interrupt is not cleared before the second timeout occurs, the system will be reset.

- Configurable timeout period.

FUNCTIONAL DESCRIPTION

Timeout

The WDT timeout can be configured by the WDT_TORR register. The minimum and maximum configurable timeout for WDT at different system clock frequencies are as follows:

System clock	2MHz	4MHz	8MHz	16MHz
Minimum timeout	64ms	32ms	16ms	8ms
Maximum timeout	32768ms	16384ms	8192ms	4096ms

Interrupt and reset

When WDT_CR [1] is set to 1, if the first timeout occurs, a WDT interrupt will be generated, and the counter will return to the corresponding count value of WDT_TORR and continue to count down. If the second timeout occurs and the WDT interrupt has not been cleared, the system will be reset.

When WDT_CR [1] is set to 0, a timeout is set. After the watchdog starts, the counter begins to decrease. When the counter reaches 0, a system reset is generated.

Write 1 to register WDT_EOI to clear the WDT interrupt signal.

CONFIGURATION FLOW

1) Enable WDT clock

The WDT clock can be enabled and disabled by RCCM.CLKENR register.

2) Configure WDT timeout output mode

The WDT timeout output mode can be configured by WDT.WDT_CR register.

3) Configure timeout period

The WDT timeout can be configured by WDT.WDT_TORR register.

4) Enable WDT interrupt

WDT interruption can be configured by WDT.WDT_CR register.

5) Enable WDT

WDT can be enabled by WDT.WDT_CR register.

6) Feed the dog regularly

When the watchdog is enabled, it is necessary to feed the watchdog within the configured timeout period by writing 0x76 to the WDT.WDT_CRR register and restarting the watchdog counter.

REGISTER LIST

Register Mapping

WDT_BASE : 0x40004000

Offset	Register Name	Description	Reset Value
0x00	WDT_CR	WDT control register	0
0x04	WDT_TORR	Timeout setting register	0
0x0C	WDT_CRR	Count restart register	0
0x10	WDT_STAT	Interrupt status register	0
0x14	WDT_EOI	Interrupt clear register	0

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write

WDT_CR: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:2	Reserved	RO	Not used	0
1	RMOD	RW	Select timeout output response 0: WDT timeout system reset 1: WDT timeout system into interrupt	0
0	WDT_EN	RW	WDT enabled. 0: Diasble 1: Enable	0

WDT_TORR: Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:4	Reserved	RO	Not used	0
3:0	TOP	RW	Select timeout: The available value range for 32-bit WDT is 2^{17} to $2^{(17+i)}$. Where i=TOP (0-15)	0

WDT_CRR: Address(0x0C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	W	Not used	0
7:0	CntRestart	W	Write 0x76 restart count	0

WDT_STAT: Address(0x10)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:1	Reserved	RO	Not used	0
0	IntrStat	RO	Interrupt status register 1: There is an interrupt generated 0: There is no interrupt generated	0

WDT_EOI: Address(0x14)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:1	Reserved	RO	Not used	0
0	IntrClr	RW	Clear interrupt flag 1:WDT Clear Interrupt Flag	0

RCC

OVERVIEW

Reset and clock controller (RCC) is used to manage the system's reset and clock, and output the state of the reset and clock.

FEATURES

- Supports reset source queries and clear operations
- Support ON/OFF and status check of clock source
- Supports 1, 2, 4, and 8 frequency division of the system clock
- Support glitch-free switching of system clock and status query of system clock

FUNCTIONAL DESCRIPTION

RESET

Reset can be divided into three categories: power reset, system reset, and peripheral reset.

- Power on reset:

When the chip is powered on normally, it will generate a power on reset, which will reset all registers.

- System reset:

The system reset all registers to their default values, except for the registers specifically designated (including SYSDIV bit of RCC_CR register, PMU_ACR register, PMU_BORCR register, PMU_BORDEG register, PMU_OTMCR register, PMU_OTFRR register, PMU_SMR register, LINM_PHY_ANR register, LINL_PHY_ANR register, AUTO_ADDRESS_EN bit and AUTO_ADDRESS_AUTOOFF and PU30K_AUTOOFF_EN bits of LINS_PHY_CR register, SYS-FWSR0 register). When the following events occur, a system reset generates:

- LDO3V3 undervoltage event occurs and reset action is selected
- VBAT overvoltage event occurs and reset action is selected
- VBAT undervoltage event occurs and reset action is selected
- Over temperature protection event occurs and reset action is selected
- Window watchdog count terminated
- Set the SOFTRSTREQ bit of the RCC CSR register

The reset source can be confirmed by querying the reset flag (RCC_CSR register). Users can clear all reset flags.

- Peripherals Reset:

The software can reset GPIO, LIN master, LIN slave, ADC, or LS_CTRL separately by configuring the peripheral reset register (RCC_PRSTR register).

CLOCK TREE

- Clock source:

AW32F020 has two built-in RC OSC clock source:

- 16MHz High Frequency RC oscillator, the output OSC16M_CLK
- 256kHz Low Frequency RC oscillator, the output OSC256K_CLK

The 16MHz RC OSC can be enabled and disabled by the HFON bit of the RCC_CR register. When enabled, until the HFRDY bit (located in the RCC_CR register) is 1, it indicates that OSC16M_CLK is ready.

The 256kHz RC OSC is always enabled and cannot be closed. LFRDY indicates whether OSC256K_CLK is ready.

- System clock:

System clock options:

- OSC16M_CLK
- LF16K_CLK is 16-divided by OSC256K_CLK

When OSC16M_CLK is selected as the system clock source, the SYSDIV bit of the RCC_CR register is used to divide the system clock.

- SSC spread spectrum:

Spead spectrum clock (SSC) is used to reduce the electromagnetic interference generated by the high-speed clock OSC16M_CLK.

REGISTER LIST

Register Detailed Description

RCC_BASE : 0x50000000

RO: Read only, W: Write only, RW: Read/Write

Clock control register (RCC_CR): Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:10	Reserved	RO	Not used	0
9	LFRDY	RO	Low frequency clock ready flag. 0: Low frequency clock not ready 1: Low frequency clock ready	1
8:6	Reserved	RO	Not used	0
5:4	SYSDIV	RW	High frequency clock division. 00: No divided 01: DIV2 10: DIV4 11: DIV8	0x2
3:2	Reserved	RO	Not used	0
1	HFRDY	RO	High frequency clock ready flag. 0: High frequency clock not ready 1: High frequency clock ready	1
0	HFON	RW	High frequency clock enable. 0: Disable 1: Enable	1

Clock configuration register (RCC_CFGR): Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:2	Reserved	RO	Not used	0
1	SWS	RO	system clock switch status. 0: LFCLK used as system clock 1: HFCLK used as system clock	1
0	SW	RW	system clock switch. 0: LFCLK selected as system clock 1: HFCLK selected as system clock	1

Peripheral reset register (RCC_PRSTR): Address(0x0C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:6	Reserved	WO	Not used	0
5	ADCRST	WO	ADC reset. 0: No effect 1: Reset ADC	0
4	LEDCTRLRST	WO	LS reset. 0: No effect 1: Reset LS	0
3	LINMRST	WO	LINM reset. 0: No effect 1: Reset LINM	0
2	LINSRST	WO	LINS reset. 0: No effect 1: Reset LINS	0
1	Reserved	RO	Not Used	0
0	GPIORST	WO	GPIO reset. 0: No effect 1: Reset GPIO	0

Peripheral clock enable register (RCC_PENR): Address(0x18)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:6	Reserved	RO	Not used	0
5	ADCEN	RW	ADC clock enable control. 0: Disable 1: Enable	0
4	LSCTRLEN	RW	LS clock enable control. 0: Disable 1: Enable	0
3	LINMEN	RW	LINM clock enable control. 0: Disable 1: Enable	0
2	LINSEN	RW	LINS clock enable control. 0: Disable 1: Enable	0
1	Reserved	RO	Not Used	0
0	GPIOEN	RW	GPIO clock enable control. 0: Disable 1: Enable	0

Control status register (RCC_CSR): Address(0x24)				
Bit	**Symbol**	**R/W**	**Description**	**Default**

31:18	Reserved	RO	Not used	0
17	UVRSTF	RO	VBAT undervoltage reset flag 0: No VBAT undervoltage reset occurred 1: VBAT undervoltage reset occurred	0
16	OVRSTF	RO	VBAT overvoltage reset flag 0: No VBAT overvoltage reset occurred 1: VBAT overvoltage reset occurred	0
15	OTPRSTF	RO	Over temperature protection reset flag. 0: No over temperature reset occurred 1: Over temperature reset occurred	0
14	WDGRSTF	RO	WWDG reset flag. 0: No WWDG reset occurred 1: WWDG reset occurred	0
13	SOFTRSTF	RO	Soft reset flag. 0: No software reset occurred 1: Software reset occurred	0
12	BOR3V3RSTF	RO	BOR3V3 reset flag. 0: No BOR3V3 reset occurred 1: BOR3V3 reset occurred	0
11	Reserved	RO	Not used	0
10	POR3V3RSTF	RO	POR3V3 reset flag. 0: No POR3V3 reset occurred 1: POR3V3 reset occurred	1
9	Reserve	RO	Not Used	0
8	CLRF	WO	Remove reset flags. 0: No effect 1: Clear reset flags	0
7:6	Reserved	RO	Not Used	0
5	SOFTRSTREQ	WO	Soft reset request. 0: No effect 1: Force a soft reset on the device	0
4:0	Reserved	RO	Not used	0

High-frequency OSC SSC register (RCC_SSCR): Address(0x40)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:15	Reserved	RO	Not used	0
14	SSCEN	RW	SSC enable. 0: Disable SSC 1: Enable SSC	0
13	Reserved	RO	Not Used	0
12:8	SSCAMP	RW	SSC amplitude.	0
7:0	SSCDIV	RW	SSC clock divider.	0x28

			Freq_ssc = freq_sys/[(SSC_DIV+1)*(SSCAMP+1)*4]	

PMU

OVERVIEW

The Power management unit (PMU) converts the input power VBAT into 3.3V and 1.5V power supplies for internal circuits, and monitors the 3.3V and 1.5V power supplies. AW32F020 provides flexible options to deal with power outages and prevent them from abnormal chip operation.

FEATURES

- Low power management.
- Integrated POR and BOR can prevent chip abnormalities caused by power supply anomalies.
- Support VBAT voltage monitoring, reset or report to CPU when undervoltage and overvoltage are generated.
- Over temperature protection.
- Support safe mode, prevent chip failure from affecting other chips.

FUNCTIONAL DESCRIPTION

Power supply

The VBAT operating voltage is 6V~18V, AW32F020 has 2 built-in LDOs. LDO3V3 normally open, can not be closed. LDO1V5、LDO3V3_SW has two different states depending on the chip operation mode:

- When AW32F020 is in RUN mode, LDO1V5 、LDO3V3_SW is in working state.
- When AW32F020 is in SLEEP mode, LDO1V5 、LDO3V3_SW can be disabled.

Security monitoring

- VBAT monitoring

AW32F020 has built-in VBAT overvoltage monitor (VBAT OV monitor) and VBAT undervoltage monitor (VBAT UV Monitor) to monitor whether the voltage on the VBAT is above or below the threshold voltage set by user.

After detecting an overvoltage or an undervoltage event, user can choose reset, interrupt, or no action.

- Power-on Reset (POR) and Brown-out Reset (BOR)
  - POR3V3:

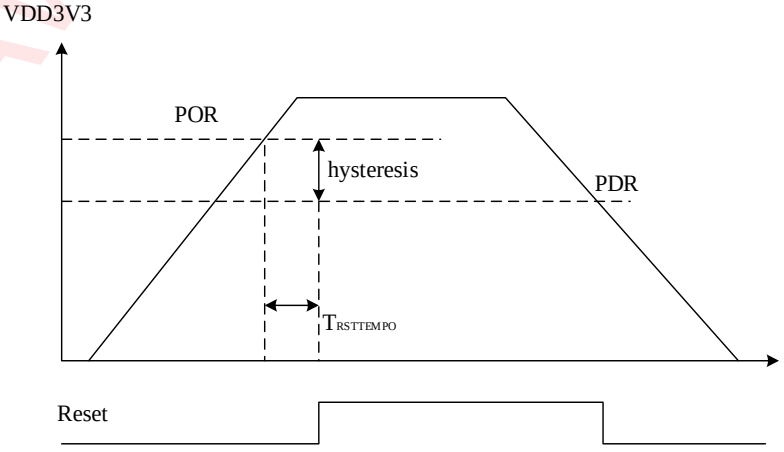


Figure 8 POR waveform

When VDD3V3 is lower than the voltage detection threshold, the system is kept in the reset state to prevent abnormal chip operation in low voltage. Refer to the ELECTRICAL CHARACTERISTICS for details of POR and BOR.

- BOR3V3:

AW32F020 has built-in Brown-out detection circuit BOR3V3 to detect whether VDD3V3 is below the set voltage threshold. The user can choose reset, interrupt, or no action.

- Over-temperature monitoring:

The OverTemp monitor monitors whether the chip temperature is above a set temperature threshold to prevent the chip from operating at abnormal temperatures. The user can choose reset, interrupt, or no action.

Low power mode

After power reset or system reset, the chip works in RUN mode. In this mode, OSC RC 16MHz is selected as the system clock source, and the power consumption can be reduced in RUN mode in the following ways:

- Reduce the system clock
- Disable the clock of unused interfaces
- Disable the CPU clock

When the CPU does not need to continue to execute the command, it can stop the CPU clock by the WFI or WFE command to save power consumption.

Sleep mode

AW32F020 can enter the sleep mode, and the power consumption of the chip is the lowest in this operating mode. Execute the following order to enter sleep mode:

1) Disable the ADC measurement
2) Disable the LS driver
3) Disable the temperature sensor
4) Configure GPIO as wake-up source if necessary
5) Configure the wake-up timer as the wake-up source if necessary
6) Configure the sleep mode register of PMU
7) CPU executes WFI and enters CPU sleep mode

Exit Sleep mode

Before sending a sleep mode request, it is necessary to ensure that at least one wake-up source is enabled, otherwise only LIN activity or POR can wake the chip from sleep mode.

After the chip wakes up from sleep mode, the software startup process is the same as the power on startup process. Sleep mode exit can be achieved through three different wake-up sources:

- When the LIN bus is at an explicit level and the time exceeds the specified value in the specification (WKUP-REC_THREE), and then switches to an implicit level, a wake-up event is immediately generated. If the sleep mode is initiated by a LIN to ground short circuit and the short circuit is subsequently eliminated, LIN will be immediately awakened. In practical applications, parallel capacitors can be reserved on the LIN bus to reduce interference on the LIN path.
- When a rising or falling edge event of GPIO is detected, a wake-up event will be immediately generated.
- By configuring the wake-up timer inside the chip, a wake-up event will be immediately generated when the preset time is reached.

Safe mode

Safe mode is a state in which the chip's functionality fails, in which the chip cannot get LIN communication or effectively control the LS. To prevent this chip from affecting other chips, all GPIO are suspended in this state.

When WDTRSTEN is valid and the WWDT count exceeds the limit, the internal WDTRSTCNT count will

increase by 1. When the chip is working properly, after WWDG reset, the CPU will set the WDTRSTCNTCLR bit and clear the WDTRSTCNT count value. If the chip malfunctions, WDTRSTCNT will continuously accumulate until it reaches the set threshold, at which point the chip will enter safe mode.

Specifically, if LIN_SW_ON is set to 1, the LIN Switch function is available. Unless powered on and reset, the safe mode cannot be exited.

REGISTER LIST

Register Detailed Description

PMU_BASE: 0x50000100

RO: Read only, W: Write only, RW: Read/Write

Control register (PMU_CR): Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:17	Reserved	RO	Not used	0
16	DWP	RW	DWP = Enable Write Access of BORCFGR, OTMCR, VBATMCR 0: Disable DWP 1: Enable DWP	0
15:9	Reserved	RO	Not Used	0
8	CWUF	WO	Clear All Wakeup Flags 0: No effect 1: Clear All Wakeup Flag	0
7:5	Reserved	RO	Not Used	0
4	CSF	WO	Clear Sleep Flag 0: No effect 1: Clear Sleep Flag	0
3	PDS	RW	Power down LDO1V5 during SLEEP mode 0: LDO1V5 Power-on During Sleep Mode 1: LDO1V5 Power-down During Sleep Mode	0
2	Reserved	RO	Not Used	0
1	HFINIT	RW	Default value is 1, Select HFCLK/LFCLK as system clock after wakeup from sleep mode 0: Select LFCLK as system clock after wakeup from sleep mode 1: Select HFCLK as system clock after wakeup from sleep mode	1
0	SLEEP	WO	Goto Sleep Mode 0: No effect 1: Prepare to Enter Sleep Mode	0

Status register (PMU_SR): Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:12	Reserved	RO	Not used	0
11	LINSWUF	RO	LINS Wakeup Flag 0: No LINS Wakeup Event Occurred	0

			1: LINS Wakeup Event Occurred	
10	Reserved	RO	Not Used	0
9	TIMWUF	RO	Wakeup Timer Flag 0: No WAKEUP_TIMER Event Occurred 1: WAKEUP_TIMER Event Occurred	0
8	GPIOWUF	RO	GPIO Wakeup Flag 0: No GPIO Wakeup Event Occurred 1: GPIO Wakeup Event Occurred	0
7:5	Reserved	RO	Not Used	0
4	SF	RO	Sleep Flag 0: No Sleep Occurred 1: Sleep Occurred	0
3:0	Reserved	RO	Not used	0

Analog control register(PMU_ACR): Address(0x0C)

Bit	Symbol	R/W	Description	Default
31:9	Reserved	RO	Not used	0
8	IBIASEN	RW	IBIAS Enable 0: Disable IBIAS 1: Enable IBIAS	0
7	Reserved	RO	Not Used	0
6	BORVSEL	RW	BOR Reference Voltage Select 0: Select GPIO5 Analog Input Voltage as BOR Reference Voltage 1: Select BG Output Voltage as BOR Reference Voltage	1
5	Reserved	RO	Not Used	0
4	BUFEN	RW	BG Buffer Enable 0: Disable BG Buffer 1: Enable BG Buffer	0
3	Reserved	RO	Not Used	0
2	TSEN	RW	Tempearture Sensor Enable 0: Disable Tempearture Sensor 1: Enable Tempearture Sensor	0
1:0	Reserved	RO	Not used	0

BOR control register(PMU_BORCR): Address(0x10)

Bit	Symbol	R/W	Description	Default
31:8	Reserved	RO	Not used	0
7:4	BOR3V3LS	RW	BOR3V3 Threshold Level Selection 0000: 2.021V 0001: 2.076V 0010: 2.134V 0011: 2.194V	0xB

			0100: 2.259V 0101: 2.327V 0110: 2.4V 0111: 2.477V 1000: 2.56V 1001: 2.648V 1010: 2.742V 1011: 2.844V 1100: 2.953V 1101: 3.071V 1110: 3.199V	
3:2	BOR3V3ACTSEL	RW	BOR3V3 Action Selection 00: No Action 01: Interrupt Generated 10: System Reset 11: No Action	1
1	Reserved	RO	Not Used	0
0	BOR3V3EN	RW	BOR3V3 Enable 0: Disable BOR3V3 1: Enable BOR3V3	0

BOR deglitch register(PMU_BORDEG): Address(0x14)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RO	Not used	0
15:11	BOR3V3NEGTHRES	RW	BOR3V3 Negative Edge Deglitch Threshold When the BOR3V3 output signal changes from 1 to 0, if the output signal keeps (BOR3V3NEGTHRES+1) continuous clock cycle (clock is 256kHz) low level, the output signal is considered as low level.	1
10:8	Reserved	RO	Not used	0
7:3	BOR3V3POSTHRES	RW	BOR3V3 Positive Edge Deglitch Threshold When the BOR3V3 output signal changes from 0 to 1, if the output signal keeps a high level (BOR3V3POSTHRES+1) clock cycle (clock is 256kHz) continuously, the output signal is considered to be high level.	0x1F
2	Reserved	RO	Not used	0
1	BOR3V3DBNC	RO	Debounced BOR3V3 Signal Output 0:BOR3V3 Signal Low after Debounced 1:BOR3V3 Signal High after Debounced	0
0	BOR3V3DEGEN	RW	BOR3V3 Deglitch Enable 0:Disable BOR3V3 Deglitch 1:Enable BOR3V3 Deglitch Note: The filter filter clock frequency is 256kHz	0

Over temperature monitor control register(PMU_OTMCR): Address(0x20)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7:4	OTLS	RW	OVTEMP Monitor Level Selection 0000: 90°C 0001: 95°C 0010: 100°C 0011: 105°C 0100: 110°C 0101: 115°C 0110: 120°C 0111: 125°C 1000: 130°C 1001: 135°C 1010: 140°C 1011: 145°C 1100: 150°C 1101: 155°C 1110: 160°C 1111: 165°C	0xb
3:2	OTACTSEL	RW	Over Temperature Action 00: No Action 01: Interrupt Generated 10: System Reset 11: No Action	0
1	Reserved	RO	Not used	0
0	OTE	RW	Over Temperature Monitor Enable 0: Disable 1: Enable	0

VBAT monitor control register(PMU_VBATMCR): Address(0x30)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:28	Reserved	RO	Not used	0
27:24	OVLS	RW	Battery Over Voltage Monitor Level Selection 0000: 20.06V 0001: 21.26V 0010: 22.58V 0011: 23.78V 0100: 25.02V 0101: 26.30V 0110: 27.54V 0111: 28.78V 1000: 30.02V 1001: 31.30V 1010: 32.46V 1011: 33.74V	0x4

			1100: 35.10V 1101: 36.30V 1110: 37.58V 1111: 38.82V	
23:22	Reserved	RO	Not used	0
21:20	OVHYS	RW	Battery Over Voltage Monitor Hysteresis Selection 00: 42mV 01: 0.714V 10: 1.344V 11: 1.624V	0x1
19	OVACTSEL	RW	Battery Over Voltage Action 0: No Action 1: System Reset	0
18:17	Reserved	RO	Not Used	0
16	OVE	RW	Battery Over Voltage Monitor Enable 0: Disable Battery Over Voltage Monitor 1: Enable Battery Over Voltage Monitor	0
15:13	Reserved	RO	Not used	0
12:8	UVLS	RW	Battery Under Voltage Monitor Level Selection 00000: 4.026V 00001: 4.124V 00010: 4.22V 00011: 4.332V 00100: 4.444V 00101: 4.64V 00110: 4.684V 00111: 4.82V 01000: 4.956V 01001: 5.1V 01010: 5.26V 01011: 5.428V 01100: 5.556V 01101: 5.7V 01110: 5.844V 01111: 5.996V 10000: 6.18V 10001: 6.356V 10010: 6.532V 10011: 6.724V 10100: 6.932V 10101: 7.148V 10110: 7.3V 10111: 7.46V 11000: 7.62V 11001: 7.796V 11010: 7.98V 11011: 8.164V 11100: 8.364V	0x9

			11101: 8.572V 11110: 8.796V 11111: 9.028V	
7:6	Reserved	RO	Not used	0
5:4	UVHYS	RW	Battery Under Voltage Monitor Hysteresis Selection 00: 16mV 01: 136mV 10: 232mV 11: 328mV	0x1
3	UVACTSEL	RW	Battery Under Voltage Action 0:No Action 1:System Reset	0
2:1	Reserved	RO	Not Used	0
0	UVE	RW	Battery Under Voltage Monitor Enable 0:Disable Battery Under Voltage Monitor 1:Enable Battery Under Voltage Monitor	0

VBAT OV debounce control register(PMU_VBATOVDCR): Address(0x34)

Bit	Symbol	R/W	Description	Default
31	OVDE	RW	Over Voltage Monitor Signal Debouncer Enable 0:Disable Over Voltage Signal Debouncer 1:Enable Over Voltage Signal Debouncer	1
30	OVDBNC	RO	Debounced Battery Over Voltage Monitor Output 0:Over Voltage Signal Low after Debounced 1:Over Voltage Signal High after Debounced	0
29:25	Reserved	RO	Not used	0
24	OVSTRB1CLK	RW	Clock Frequency Strobling 1 Select for Debouncing 0:Select High Frequency Clock for Debouncing 1:Select Low Frequency Clock for Debouncing	0
23:16	OVTHRES1	RW	Over Voltage Debouncing Threshold for 0to1 Transition When the overvoltage signal changes from 0 to 1, the overvoltage signal is considered valid if the overvoltage signal is kept high for (OVTHRES1+1) consecutive clock cycles (the clock is selected by OVSTRB1CLK)	0x1
15:9	Reserved	RO	Not used	0
8	OVSTRB0CLK	RW	Low Frequency Strobling 0 Select for Debouncing 0:Select High Frequency Clock for Debouncing 1:Select Low Frequency Clock for Debouncing	1
7:0	OVTHRES0	RW	Over Voltage Debouncing Threshold for 1to0 Transition When the overvoltage signal changes from 1 to 0, the overvoltage signal is considered valid if the overvoltage signal keeps a high level for (OVTHRES0+1) consecutive clock cycles (the clock is selected by OVSTRB0CLK)	0xFF

VBAT UV debounce control register(PMU_VBATUVDCR): Address(0x38)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31	UVDE	RW	Under Voltage Monitor Signal Debouncer Enable 0:Disable Under Voltage Signal Debouncer 1:Enable Under Voltage Signal Debouncer	1
30	UVDBNC	RO	Debounced Battery Under Voltage Monitor Output 0:Under Voltage Signal Low after Debounced 1:Under Voltage Signal High after Debounced	0
29:25	Reserved	RO	Not Used	0
24	UVSTRB1CLK	RW	Low Frequency Strobling 1 Select for Debouncing 0:Select High Frequency Clock for Debouncing 1:Select Low Frequency Clock for Debouncing	0
23:16	UVTHRES1	RW	Under Voltage Debouncing Threshold for 0 to1 Transition Configured by software. When the undervoltage signal changes from 0 to 1, it is considered effective if the undervoltage signal keeps high level for (VTHRES1+1) consecutive clock cycles (the clock is selected by UVSTRB1CLK)	1
15:9	Reserved	RO	Not used	0
8	UVSTRB0CLK	RW	Low Frequency Strobling 0 Select for Debouncing 0:Select High Frequency Clock for Debouncing 1:Select Low Frequency Clock for Debouncing	1
7:0	UVTHRES0	RW	Under Voltage Debouncing Threshold for 1 to 0 Transition When the undervoltage signal changes from 1 to 0, if the undervoltage signal keeps a high level for (VTHRES0+1) consecutive clock cycles (the clock is selected by UVSTRB0CLK), the undervoltage signal is considered effective	0xFF

Interrupt control/status register(PMU_INTCSR): Address(0x40)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:28	Reserved	RO	Not used	0
27	BORC	WO	BOR1V5 and BOR3V3 Interrupt Clear 1:Clear BOR1V5 and BOR3V3 Interrupt Flag	0
26	Reserved	RO	Not Used	0
25	BOR3V3F	RO	BOR3V3 Interrupt Flag 0:No BOR3V3 Interrupt Generated 1:BOR3V3 Interrupt Generated	0
24:20	Reserved	RO	Not used	0
19	OTC	WO	Over Temperature Interrupt Clear 1:Clear Over Temperature Interrupt Flag	0
18:17	Reserved	RO	Not Used	0
16	OTF	RO	Over Temperature Interrupt Flag 0:No Over Temperature Interrupt Generated 1:Over Temperature Interrupt Generated	0

15	OVPOL	RW	Battery Over Voltage Monitor OV Interrupt Polarity 0:Disable Flip Polarity 1:Enable Flip Polarity	0
14:12	Reserved	RO	Not used	0
11	OVC	WO	Over Voltage Interrupt Clear 1:Clear Over Voltage Interrupt Flag	0
10	OVIE	RW	Over Voltage Interrupt Enable 0:Disable Over Voltage Interrupt 1:Enable Over Voltage Interrupt	0
9	Reserved	RO	Not used	0
8	OVF	RO	Over Voltage Interrupt Flag When the VBAT exceeds the set threshold voltage and the OVIE is valid, the bit is automatically set by the hardware, and the write OVC can clear the bit 0:No Over Voltage Interrupt Generated 1:Over Voltage Interrupt Generated	0
7	UVPOL	RW	Battery Under Voltage Monitor OV Interrupt Polarity When the bit is 1, the VBAT undervoltage signal is 0 and the UVIE is 1, the undervoltage interrupt is considered. When this bit is 0, the VBAT undervoltage signal is 1 and the UVIE is 1, the undervoltage interrupt is considered to be generated 0:Disable Flip Polarity 1:Enable Flip Polarity	0
6:4	Reserved	RO	Not used	0
3	UVC	WO	Under Voltage Interrupt Clear 1:Clear Under Voltage Interrupt Flag	0
2	UVIE	RW	Under Voltage Interrupt Enable 0:Disable Under Voltage Interrupt 1:Enable Under Voltage Interrupt	0
1	Reserved	RO	Not used	0
0	UVF	RO	Under Voltage Interrupt Flag When VBAT is below the set threshold voltage and UVIE is valid, the bit is automatically set by the hardware, and the bit can be cleared by writing UVC 0:No Under Voltage Interrupt Generated 1:Under Voltage Interrupt Generated	0

Safe mode register(PMU_SMR): Address(0x60)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:24	SMRPASSWD	WO	Safe Mode Register Password To write a different bit in the safe mode register, the bit should be written to 0x5C. Otherwise invalid This bit is always read out to be 0	0
23:17	Reserved	RO	Not used	0

16	SAFEMODEREQ	RW	If Password is valid, Safe Mode Request 0: No effect 1:Enter Safe Mode	0
15:13	Reserved	RO	Not used	0
12	WDTRSTCNTEN	RW	If Password is valid, Watchdog Reset Counter Enable 0:Disable Watchdog Reset Counter 1:Enable Watchdog Reset Counter	0
11:9	Reserved	RO	Not Used	0
8	WDTRSTCNTCLR	WO	If Password is valid, Watchdog Reset Counter Clear 0: No effect 1:Clear Watchdog Reset Counter	0
7	Reserved	RO	Not Used	0
6:4	WDTRSTCNT	RO	Watchdog Reset Counter Current Value	0x0
3	Reserved	RO	Not Used	0
2:0	WDTRSTTHRES	RW	Watchdog Reset Counter Threshold When Watchdog Reset Counter reaches threshold, the device goes to safe mode	0x5

SYSCTRL

OVERVIEW

The SYSCTRL module is mainly used to store calibration values. These calibration values are used for the adjustment of the simulation part.

FEATURES

- Supports two backup registers
- Support calibration register configuration

FUNCTIONAL DESCRIPTION

Back up register

This module contains two firmware scratches SYS_FWSR0 and SYS_FWSR1. SYS_FWSR0 is not subject to various software and hardware reset, only by power down reset control; SYS_FWSR1 can be reset by various software and hardware.

TRIM write configuration

To configure TRIM, follow the following process:

1) Enable TRIM write mode and write 0xE to SYS_TRIMENR;
2) Write TRIM values to the corresponding TRIM registers;
3) Disable TRIM write mode and write 0x0 to SYS_TRIMENR;
4) To latch the value of the TRIM register, write 0x10 to SYS_TRIMENR.

REGISTER LIST

Register Mapping

SYSCTRL_BASE : 0x50000900

Offset	Register Name	Description	Reset Value
0x00	SYS_FWSR0	firmware scratch register0.	0x00
0x04	SYS_FWSR1	firmware scratch register1.	0x00
0x08	SYS_TRIMENR	trim enable register.	0x00
0x0c	SYS_TRIMSTR	trim status register.	0x00
0x10	SYS_PMUBGTRIMR	pmu bg trim register.	0x00
0x14	SYS_PMUIBIASTRIMR	pmu bias trim register.	0x00
0x1c	SYS_PMULDO1V5TRIMR	pmu ldo 1.5V trim register.	0x00
0x20	SYS_PMULDO3V3TRIMR	pmu ldo 3.3V trim register.	0x00
0x24	SYS_PMUILSTRIMR	PMU ILS TRIM register.	0x00
0x28	SYS_PMUVFWBIASTRIMR	PMU VFW BIAS TRIM register.	0x00
0x30	SYS_OSC256KTRIMR	low frequent osc trim register.	0x80
0x34	SYS_OSC16MTRIMR	high frequent osc trim register.	0x00
0x40	SYS_LSBR0TRIMR	LS br register. step 120uA	0x00
0x44	SYS_LSBR1TRIMR	LS br register. step 120uA	0x00
0x48	SYS_LSBR2TRIMR	LS br register. step 120uA	0x00
0x50	SYS_LSVFWTRIMR	LS vfw detect trim register.	0x00

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write

SYS_FWSR0: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:0	FWSR0	RW	Firmware scratch register 0. Only reset at power-on (e.g contents retained in Hibernate mode and retained despite any hard or soft resets).	0x0

SYS_FWSR1: Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:0	FWSR1	RW	Firmware scratch register 1. Only reset at power-on (e.g contents retained in Hibernate mode and retained despite any hard or soft resets).	0x0

SYS_TRIMENR: Address(0x08)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:5	Reserved	RO	Not used	0
4	TRIM_LOCK	RW	Lock. b0:UNLOCK b1:LOCK	0
3:0	TRIM_ACCESS_KEY	RW	Access key. b0000:KEY_NOTCODE b1110:KEY_CODE	0x0

SYS_TRIMSTR: Address(0x0C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:1	Reserved	RO	Not used	0
0	TRIMSTR	RO	Trim status. b0:Ineffect b1:Effect	0x0

SYS_PMUBGTRIMR: Address(0x10)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:5	Reserved	RO	Not used	0
4:0	PMUBG_TRIM	RW	Bandgap trim.	0x0

SYS_PMUIBIASTRIM: Address(0x14)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:5	Reserved	RO	Not used	0
4:0	PMUIBIAS_TRIM	RW	Bias trim.	0x00

SYS_PMULDO1V5TRIMR: Address(0x1C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:4	Reserved	RO	Not used	0
3:0	PMULDO1V5_TRIM	RW	Value.	0x0

SYS_PMULDO3V3TRIMR: Address(0x20)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:4	Reserved	RO	Not used	0
3:0	PMULDO3V3_TRIM	RW	Value.	0x0

SYS_PMUILSTRIMR: Address(0x24)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:2	Reserved	RO	Not used	0
1:0	PMUILS_TRIM	RW	IBIAS provides TRIM signal with 30μA current for LS0 to 3	0x0

SYS_PMUVFWBIASTRIMR: Address(0x28)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:2	Reserved	RO	Not used	0
1:0	PMUVFWBIAS_TRIM	RW	The IBIAS provides the TRIM signal of 2.5μA current to the VFW	0x0

SYS_OSC256KTRIMR: Address(0x30)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7:0	OSC256K_TRIM	RW	Low frequency trim.	0x80

SYS_OSC16MTRIMR: Address(0x34)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7:0	OSC16M_TRIM	RW	High frequency trim.	0x0

SYS_LSBR0TRIMR: Address(0x40)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RO	Not used	0
8:0	LSBR0_TRIM	RW	LSBR trim, 120μA per step.	0x0

SYS_LSBR1TRIMR: Address(0x44)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RO	Not used	0
8:0	LSBR1_TRIM	RW	LSBR trim, 120μA per step.	0x0

SYS_LSBR2TRIMR: Address(0x48)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RO	Not used	0
8:0	LSBR2_TRIM	RW	LSBR trim, 120μA per step.	0x0

SYS_LSVFWTRIMR: Address(0x50)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7:0	VFW_TRIM	RW	Trim value, 10μA per step.	0x0

UART

OVERVIEW

UART can flexibly exchange full duplex data with external devices, meeting the requirements of external devices for the industrial standard NRZ asynchronous serial data format. UART implements multiple baud rates through a fractional baud rate generator.

UART not only supports synchronous unidirectional communication and half duplex single wire communication, but also LIN and IrDA.

FEATURES

- Full-duplex asynchronous communication
- Baud rate generator system
- Universal programmable transmit and receive baud rate
- Programmable data word length (7-bit, 8-bit or 9-bit)
- Configurable stop bits (supporting 1-bit, 1.5-bit or 2-bit)
- Communication control/error detection flags
- Parity control
  - Send parity bit
  - Check the parity of received digital bytes
- Interrupt sources with flags
- Support for multi-processor communication

FUNCTIONAL DESCRIPTION

UART

UART bidirectional communication requires at least two pins: receiver pin (RX) and transmitter pin (TX):

- RX (receiver pin)

RX is a serial data input pin. Using oversampling techniques for data recovery, which can distinguish valid input data and noise.

- TX (transmitter pin)

If the transmitter is turned off, the output pin mode is determined by its I/O port configuration. If the transmitter is enabled but there is no data to be transmitted, the TX pin is at a high level.

CONFIGURE FLOW

Transmitter configuration

1. Set the M bit of USART_CR1 to define the word length.
2. Set the desired baud rate by the USART_BRR register.
3. Set the number of stop bits in USART_CR2.
4. Enable USART by setting the UE in the USART_CR1 register to 1.
5. If using multi buffer communication, configure the DMA enable bit (DMAT) in USART_CR3. Configure DMA registers as described in multi buffer communication.
6. Set the TE bit in USART_CR1 and send an idle frame as the first data transmission.
7. Write the data to be sent into the USART_TDR register (this action will clear the TXE bit). Repeat this step 7 for each data to be sent when there is only one buffer. Before repeating step 7, wait for TXE to become 1.
8. After writing the last data word in the USART_TDR register, wait for TC=1, which indicates the end of the transmission of the last data frame. Before shutting down USART or entering shutdown mode, it is necessary to confirm the end of the transmission to avoid damaging the last transmission.

Receiver configuration

1. Set the M bit of USART_CR1 to define the word length.
2. Set the desired baud rate by the USART_BRR register.
3. Set the number of stop bits in USART_CR2.
4. Activate USART by setting the UE in the USART_CR1 register to 1.
5. If using multi buffer communication, configure the DMA enable bit (DMAR) in USART_CR3. Configure DMA registers as described in multi buffer communication.
6. Set the RE of USART_CR1 to 1 to activate the receiver and start searching for the starting bit.
7.

REGISTER LIST

Register Mapping

UART Base: 0x40004B00

Offset	Register Name	Description	Reset Value
0x00	USART_SR	USART state register	0x0
0x04	USART_DR	USART data register	0x0
0x08	USART_BRR	USART baud rate register	0x0
0x0c	USART_CR1	USART control register 1	0x0
0x10	USART_CR2	USART control register 2	0x60
0x14	USART_CR3	USART control register 3	0x0

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write, RO/W1: Read only, Write 1 clear

USART_SR: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	R	Not used	0
8	LBD	R	LBD: LIN break detection flag 0: LIN Break not detected 1: LIN break detected	0
7	TXE	R	TXE: Transmit data register empty 0: Data is not transferred to the shift register 1: Data is transferred to the shift register	1
6	TC	R	TC: Transmission complete 0: Transmission is not complete 1: Transmission is complete	1
5	RXNE	R	RXNE: Read data register not empty 0: Data is not received 1: Received data is ready to be read.	0
4	IDLE	R	IDLE: IDLE line detected 0: No Idle Line is detected 1: Idle Line is detected	0
3	ORE	R	ORE: Overrun error 0: No Overrun error 1: Overrun error is detected	0
2	NE	R	NE: Noise error flag 0: No noise is detected 1: Noise is detected	0
1	FE	R	FE: Framing error 0: No Framing error is detected 1: Framing error or break character is detected	0
0	PE	R	PE: Parity error 0: No parity error 1: Parity error	0

USART_DR: Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RW	Not used	0
8:0	DR[8:0]	RW	DR[8:0]: Data value	0

USART_BRR: Address(0x08)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	Reserved	RW	Not used	0
15:4	DIV_Mantissa[11:0]	RW	DIV_Mantissa[11:0]: mantissa of USARTDIV These 12 bits define the mantissa of the USART Divider (USARTDIV)	0
3:0	DIV_Fraction[3:0]	RW	DIV_Fraction[3:0]: fraction of USARTDIV These 4 bits define the fraction of the USART	0

			Divider (USARTDIV)	

USART_CR1: Address(0x0C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:14	Reserved	RW	Not used	0
13	UE	RW	UE: USART enable 0: USART prescaler and outputs disabled 1: USART enabled	0
12	M	RW	M: Word length 0: 1 Start bit, 8 Data bits, n Stop bit 1: 1 Start bit, 9 Data bits, n Stop bit Note: The M bit must not be modified during a data transfer (both transmission and reception)	0
11	WAKE	RW	WAKE: Wakeup method 0: Idle Line 1: Address Mark	0
10	PCE	RW	PCE: Parity control enable 0: Parity control disabled 1: Parity control enabled	0
9	PS	RW	PS: Parity selection 0: Even parity 1: Odd parity	0
8	PEIE	RW	PEIE: PE interrupt enable 0: Interrupt is inhibited 1: A USART interrupt is generated whenever PE=1 in the USART_SR register	0
7	TXEIE	RW	TXEIE: TXE interrupt enable 0: Interrupt is inhibited 1: A USART interrupt is generated whenever TXE=1 in the USART_SR register	0
6	TCIE	RW	TCIE: Transmission complete interrupt enable 0: Interrupt is inhibited 1: A USART interrupt is generated whenever TC=1 in the USART_SR register	0
5	RXNEIE	RW	RXNEIE: RXNE interrupt enable 0: Interrupt is inhibited 1: A USART interrupt is generated whenever ORE=1 or RXNE=1 in the USART_SR register	0
4	IDLEIE	RW	IDLEIE: IDLE interrupt enable 0: Interrupt is inhibited 1: A USART interrupt is generated whenever IDLE=1 in the USART_SR register	0
3	TE	RW	TE: Transmitter enable 0: Transmitter is disabled 1: Transmitter is enabled	0
2	RE	RW	RE: Receiver enable	0

			0: Receiver is disabled 1: Receiver is enabled and begins searching for a start bit	
1	RWU	RW	RWU: Receiver wakeup 0: Receiver in active mode 1: Receiver in mute mode	0
0	SBK	RW	SBK: Send break 0: No break character is transmitted 1: Break character will be transmitted	0

USART_CR2: Address(0x10)

Bit	Symbol	R/W	Description	Default
31:15	Reserved	RW	Not used	0
14	LINEN	RW	LINEN: LIN mode enable 0: LIN mode disabled 1: LIN mode enabled The LIN mode enables the capability to send LIN Synch Breaks (13 low bits) using the SBK bit in the USART_CR1 register, and to detect LIN Sync breaks.	0
13:12	STOP	RW	STOP: STOP bits 00: 1 Stop bit 01: 0.5 Stop bit 10: 2 Stop bits 11: 1.5 Stop bit	0
11:7	Reserved	RW	Not used	0
6	LBDIE	RW	LBDIE: LIN break detection interrupt enable 0: Interrupt is inhibited 1: An interrupt is generated whenever LBD=1 in the USART_SR register	0
5	LBDL	RW	LBDL: lin break detection length 0: 10 bit break detection 1: 11 bit break detection	0
4	Reserved	RW	Not used	0
3:0	ADD[3:0]	RW	ADD[3:0]: Address of the USART node This is used in multiprocessor communication during mute mode, for wake up with address mark detection.	0

USART_CR3: Address(0x14)

Bit	Symbol	R/W	Description	Default
31:4	Reserved	RW	Not used	0
3	HDSEL	RW	HDSEL: Half-duplex selection 0: Half duplex mode is not selected 1: Half duplex mode is selected	0
2	IRLP	RW	IRLP: IrDA low-power	0

			0: Normal mode 1: Low-power mode	
1	IREN	RW	IREN: IrDA mode enable 0: IrDA disabled 1: IrDA enabled	0
0	EIE	RW	EIE: Error interrupt enable 0: Interrupt is inhibited 1: An interrupt is generated whenever DMAR=1 in the USART_CR3 register and FE=1 or ORE=1 or NE=1 in the USART_SR register.	0

GPIO

OVERVIEW

The GPIO module has a total of 8 general purpose I/O pins. Support pull up and down selection, data input and output direction control function. Each IO port supports dual edge interrupt trigger and clear function. The GPIO module has separate gated clock, reset for the entire module with separate clock and reset.

FEATURES

- 8 programmable input/output interfaces.
- Each I/O port supports weak pull-up and weak pull-down operations.
- Data input and output direction control.
- Each I/O port supports dual edge triggering, clearing, and status querying of interrupts.

FUNCTIONAL DESCRIPTION

I/O functions

GPIO can control the direction of data input and output for each IO by input enable and output enable.

Input function: Configure the corresponding port to GPIO mode by port reuse, enable the input of the port, and read the level status of each IO port by the GPIO_IDR register.

Output function: Configure the corresponding port to GPIO mode by port reuse, enable the output of the port, and configure the output data of each port by the GPIO_ODR register.

Weak pull-up/down function

Port reuse is used to configure the corresponding port to GPIO mode. By enabling the weak pull up or weak pull down of the port, users can achieve a weak pull up or weak pull down operation for each IO port. The weak pull up and weak pull down enable control signals of each IO port are independent of each other, and users can configure the weak pull up and weak pull down operation of any IO according to the actual use situation.

Wakeup function in sleep mode

Once the port is configured into GPIO mode by port multiplexing, it goes into sleep mode, and any level change of the GPIO port will generate a wakeup event, which wakes up the whole system.

After the GPIO interrupt is enabled, the rising edge and falling edge of the port can be enabled according to

the actual application requirements. Event wakeup is triggered when the port produces the expected level change.

The configurations of each IO are independent of each other, as are the rising edge enablement and falling edge enablement of each IO.

Triggering and clearing interrupts

The port is multiplexed into GPIO mode to enable gated clock and input enable. The interrupt enable, rising edge enable, or falling edge enable of a port triggers an interrupt when the port produces the expected level change. The interrupt source can be queried by reading the interrupt status register GPIO_IRQPR. Each interrupt generated by an IO port can be cleared by writing 1 to the bit corresponding to the interrupt status register port separately.

CONFIGURE FLOW

GPIO input function configuration

- Enable the GPIO clock:

The GPIO clock can be enabled and disabled by the RCC.RCC_PENR register.

- Multiplexing to GPIO mode:

The mode of a port can be configured by the AFIO.AFIO_MFPR register.

- Enable input:

The input enable for ports GPIO0 to GPIO5 can be configured by the GPIO.GPIO_IBER register.

- Configure pull-down:

Pull-down resistors for ports GPIO0 to GPIO5 can be configured by the GPIO.GPIO_PUPDR register.

- Get the input pin status:

The level status of the input pin can be obtained by reading the GPIO.GPIO_IDR register.

GPIO output function configuration

- Enable the GPIO clock:

The GPIO clock can be enabled and disabled by the RCC.RCC_PENR register.

- Multiplexing to GPIO mode:

The mode of a port can be configured by the AFIO.AFIO_MFPR register.

- Output enable:

The output enable from ports GPIO0 to GPIO5 can be configured by the GPIO.GPIO_OBER register

- Configure the output data:

Output data from ports GPIO0 to GPIO5 can be configured by the GPIO.GPIO_ODR register.

GPIO interrupt configuration

- Enable the GPIO clock:

The GPIO clock can be enabled and disabled by the RCC.RCC_PENR register.

- Multiplexing to GPIO mode:

The mode of a port can be configured by the AFIO.AFIO_MFPR register.

- Enable input

The input enable for ports GPIO0 to GPIO5 can be configured by the GPIO.GPIO_IBER register.

- Enable GPIO edge detection:

Falling edge detection for ports GPIO0 to GPIO5 enables configuration by the GPIO.GPIO_FTSR register.

Rising edge detection for ports GPIO0 to GPIO5 enables configuration by the GPIO.GPIO_RTSR register.

- Enable GPIO port interrupt:

The interrupt enable on ports GPIO0 to GPIO5 is configured by the GPIO.GPIO_IRQENR register.

- Get the GPIO interrupt status:

A GPIO interrupt is triggered when the port level status of a port configured for GPIO interrupt mode changes as expected. When a GPIO interrupt is triggered, the interrupt status of the port can be obtained by reading the GPIO.GPIO_IRQPR register.

- Clear the GPIO interrupt:

The interrupt state of a port can be cleared by writing 1 to the corresponding interrupt state register bit.

REGISTER LIST

Register Mapping

GPIO_BASE: 0x50000A00

Offset	Register Name	Description	Reset Value
0x00	GPIO_IBER	GPIO Input Enable Register	0x00
0x04	GPIO_OBER	GPIO Output Enable Register	0x00
0x08	GPIO_IDR	GPIO Input Data Register	0x00
0x0C	GPIO_ODR	GPIO Output Data Register	0x00
0x10	GPIO_PUPDR	GPIO Pull-Up/Pull-Down Register	0x00
0x14	GPIO_IRQER	GPIO Interrupt Request Enable Register	0x00
0x18	GPIO_FTSR	GPIO Falling Edge Detection Enable Register	0x00
0x1C	GPIO_RTSR	GPIO Rising Edge Detection Enable Register	0x00
0x20	GPIO_IRQPR	GPIO Interrupt Status Register	0x00

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write, RO/W1: Read only, Write 1 clear

GPIO_IBER: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7	IBER7	RW	GPIO7 Input Enable Bit 0: Disable GPIO7 Input 1: Enable GPIO7 Input	0
6	IBER6	RW	GPIO6 Input Enable Bit 0: Disable GPIO6 Input 1: Enable GPIO6 Input	0
5	IBER5	RW	GPIO5 Input Enable Bit 0: Disable GPIO5 Input 1: Enable GPIO5 Input	0
4	IBER4	RW	GPIO4 Input Enable Bit 0: Disable GPIO4 Input 1: Enable GPIO4 Input	0
3	IBER3	RW	GPIO3 Input Enable Bit	0

			0: Disable GPIO3 Input 1: Enable GPIO3 Input	
2	IBER2	RW	GPIO2 Input Enable Bit 0: Disable GPIO2 Input 1: Enable GPIO2 Input	0
1	IBER1	RW	GPIO1 Input Enable Bit 0: Disable GPIO1 Input 1: Enable GPIO1 Input	0
0	IBER0	RW	GPIO0 Input Enable Bit 0: Disable GPIO0 Input 1: Enable GPIO0 Input	0

GPIO_OBER: Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
5	OBER7	RW	GPIO7 Output Enable Bit 0: Disable GPIO7 Output 1: Enable GPIO7 Output	0
5	OBER6	RW	GPIO6 Output Enable Bit 0: Disable GPIO6 Output 1: Enable GPIO6 Output	0
5	OBER5	RW	GPIO5 Output Enable Bit 0: Disable GPIO5 Output 1: Enable GPIO5 Output	0
4	OBER4	RW	GPIO4 Output Enable Bit 0: Disable GPIO4 Output 1: Enable GPIO4 Output	0
3	OBER3	RW	GPIO3 Output Enable Bit 0: Disable GPIO3 Output 1: Enable GPIO3 Output	0
2	OBER2	RW	GPIO2 Output Enable Bit 0: Disable GPIO2 Output 1: Enable GPIO2 Output	0
1	OBER1	RW	GPIO1 Output Enable Bit 0: Disable GPIO1 Output 1: Enable GPIO1 Output	0
0	OBER0	RW	GPIO0 Output Enable Bit 0: Disable GPIO0 Output 1: Enable GPIO0 Output	0

GPIO_IDR: Address(0x08)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0

7	IDR7	RO	GPIO7 Input Data Bit 0: GPIO7 Input Data Value = 0 1: GPIO7 Input Data Value = 1	0
6	IDR6	RO	GPIO6 Input Data Bit 0: GPIO6 Input Data Value = 0 1: GPIO6 Input Data Value = 1	0
5	IDR5	RO	GPIO5 Input Data Bit 0: GPIO5 Input Data Value = 0 1: GPIO5 Input Data Value = 1	0
4	IDR4	RO	GPIO4 Input Data Bit 0: GPIO4 Input Data Value = 0 1: GPIO4 Input Data Value = 1	0
3	IDR3	RO	GPIO3 Input Data Bit 0: GPIO3 Input Data Value = 0 1: GPIO3 Input Data Value = 1	0
2	IDR2	RO	GPIO2 Input Data Bit 0: GPIO2 Input Data Value = 0 1: GPIO2 Input Data Value = 1	0
1	IDR1	RO	GPIO1 Input Data Bit 0: GPIO1 Input Data Value = 0 1: GPIO1 Input Data Value = 1	0
0	IDR0	RO	GPIO0 Input Data Bit 0: GPIO0 Input Data Value = 0 1: GPIO0 Input Data Value = 1	0

GPIO_ODR: Address(0x0C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7	ODR7	RW	GPIO7 Output Data Bit 0: GPIO7 Output Data Value = 0 1: GPIO7 Output Data Value = 1	0
6	ODR6	RW	GPIO6 Output Data Bit 0: GPIO6 Output Data Value = 0 1: GPIO6 Output Data Value = 1	0
5	ODR5	RW	GPIO5 Output Data Bit 0: GPIO5 Output Data Value = 0 1: GPIO5 Output Data Value = 1	0
4	ODR4	RW	GPIO4 Output Data Bit 0: GPIO4 Output Data Value = 0 1: GPIO4 Output Data Value = 1	0
3	ODR3	RW	GPIO3 Output Data Bit 0: GPIO3 Output Data Value = 0 1: GPIO3 Output Data Value = 1	0
2	ODR2	RW	GPIO2 Output Data Bit	0

			0: GPIO2 Output Data Value = 0 1: GPIO2 Output Data Value = 1	
1	ODR1	RW	GPIO1 Output Data Bit 0: GPIO1 Output Data Value = 0 1: GPIO1 Output Data Value = 1	0
0	ODR0	RW	GPIO0 Output Data Bit 0: GPIO0 Output Data Value = 0 1: GPIO0 Output Data Value = 1	0

GPIO_PUPDR: Address(0x10)

Bit	Symbol	R/W	Description	Default
31:16	Reserved	RO	Not used	0
15:14	PUPDR7	RW	GPIO7 Pull-Up/Pull-Down Control 00: GPIO7 No Pull-Up/Pull-Down 01: GPIO7 Pull-Up Mode 10: GPIO7 Pull-Down Mode 11: Reserved	0
13:12	PUPDR6	RW	GPIO6 Pull-Up/Pull-Down Control 00: GPIO6 No Pull-Up/Pull-Down 01: GPIO6 Pull-Up Mode 10: GPIO6 Pull-Down Mode 11: Reserved	0
11:10	PUPDR5	RW	GPIO5 Pull-Up/Pull-Down Control 00: GPIO5 No Pull-Up/Pull-Down 01: GPIO5 Pull-Up Mode 10: GPIO5 Pull-Down Mode 11: Reserved	0
9:8	PUPDR4	RW	GPIO4 Pull-Up/Pull-Down Control 00: GPIO4 No Pull-Up/Pull-Down 01: GPIO4 Pull-Up Mode 10: GPIO4 Pull-Down Mode 11: Reserved	0
7:6	PUPDR3	RW	GPIO3 Pull-Up/Pull-Down Control 00: GPIO3 No Pull-Up/Pull-Down 01: GPIO3 Pull-Up Mode 10: GPIO3 Pull-Down Mode 11: Reserved	0
5:4	PUPDR2	RW	GPIO2 Pull-Up/Pull-Down Control 00: GPIO2 No Pull-Up/Pull-Down 01: GPIO2 Pull-Up Mode 10: GPIO2 Pull-Down Mode 11: Reserved	0
3:2	PUPDR1	RW	GPIO1 Pull-Up/Pull-Down Control 00: GPIO1 No Pull-Up/Pull-Down 01: GPIO1 Pull-Up Mode 10: GPIO1 Pull-Down Mode	0

			11: Reserved	
1:0	PUPDR0	RW	GPIO0 Pull-Up/Pull-Down Control 00: GPIO0 No Pull-Up/Pull-Down 01: GPIO0 Pull-Up Mode 10: GPIO0 Pull-Down Mode 11: Reserved	0

GPIO_IRQENR: Address(0x14)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7	IRQENR7	RW	GPIO7 Interrupt Enable Bit 0: Disable GPIO7 Interrupt 1: Enable GPIO7 Interrupt	0
6	IRQENR6	RW	GPIO6 Interrupt Enable Bit 0: Disable GPIO6 Interrupt 1: Enable GPIO6 Interrupt	0
5	IRQENR5	RW	GPIO5 Interrupt Enable Bit 0: Disable GPIO5 Interrupt 1: Enable GPIO5 Interrupt	0
4	IRQENR4	RW	GPIO4 Interrupt Enable Bit 0: Disable GPIO4 Interrupt 1: Enable GPIO4 Interrupt	0
3	IRQENR3	RW	GPIO3 Interrupt Enable Bit 0: Disable GPIO3 Interrupt 1: Enable GPIO3 Interrupt	0
2	IRQENR2	RW	GPIO2 Interrupt Enable Bit 0: DisableGPIO2 Interrupt 1: Enable GPIO2 Interrupt	0
1	IRQENR1	RW	GPIO1 Interrupt Enable Bit 0: Disable GPIO1 Interrupt 1: Enable GPIO1 Interrupt	0
0	IRQENR0	RW	GPIO0 Interrupt Enable Bit 0: Disable GPIO0 Interrupt 1: Enable GPIO0 Interrupt	0

GPIO_FTSR: Address(0x18)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7	FTSR7	RW	GPIO7 Falling Edge Detection Enable Bit 0: Disable GPIO7 Falling Edge Detection 1: Enable GPIO7 Falling Edge Detection	0

6	FTSR6	RW	GPIO6 Falling Edge Detection Enable Bit 0: Disable GPIO6 Falling Edge Detection 1: Enable GPIO6 Falling Edge Detection	0
5	FTSR5	RW	GPIO5 Falling Edge Detection Enable Bit 0: Disable GPIO5 Falling Edge Detection 1: Enable GPIO5 Falling Edge Detection	0
4	FTSR4	RW	GPIO4 Falling Edge Detection Enable Bit 0: Disable GPIO4 Falling Edge Detection 1: Enable GPIO4 Falling Edge Detection	0
3	FTSR3	RW	GPIO3 Falling Edge Detection Enable Bit 0: Disable GPIO3 Falling Edge Detection 1: Enable GPIO3 Falling Edge Detection	0
2	FTSR2	RW	GPIO2 Falling Edge Detection Enable Bit 0: Disable GPIO2 Falling Edge Detection 1: Enable GPIO2 Falling Edge Detection	0
1	FTSR1	RW	GPIO1 Falling Edge Detection Enable Bit 0: Disable GPIO1 Falling Edge Detection 1: Enable GPIO1 Falling Edge Detection	0
0	FTSR0	RW	GPIO0 Falling Edge Detection Enable Bit 0: Disable GPIO0 Falling Edge Detection 1: Enable GPIO0 Falling Edge Detection	0

GPIO_RTSR: Address(0x1C)

Bit	Symbol	R/W	Description	Default
31:8	Reserved	RO	Not used	0
7	RTSR7	RW	GPIO7 Rising Edge Detection Enable Bit 0: Disable GPIO7 Rising Edge Detection 1: Enable GPIO7 Rising Edge Detection	0
6	RTSR6	RW	GPIO6 Rising Edge Detection Enable Bit 0: Disable GPIO6 Rising Edge Detection 1: Enable GPIO6 Rising Edge Detection	0
5	RTSR5	RW	GPIO5 Rising Edge Detection Enable Bit 0: Disable GPIO5 Rising Edge Detection 1: Enable GPIO5 Rising Edge Detection	0
4	RTSR4	RW	GPIO4 Rising Edge Detection Enable Bit 0: Disable GPIO4 Rising Edge Detection 1: Enable GPIO4 Rising Edge Detection	0
3	RTSR3	RW	GPIO3 Rising Edge Detection Enable Bit 0: Disable GPIO3 Rising Edge Detection 1: Enable GPIO3 Rising Edge Detection	0
2	RTSR2	RW	GPIO2 Rising Edge Detection Enable Bit 0: Disable GPIO2 Rising Edge Detection 1: Enable GPIO2 Rising Edge Detection	0

1	RTSR1	RW	GPIO1 Rising Edge Detection Enable Bit 0: Disable GPIO1 Rising Edge Detection 1: Enable GPIO1 Rising Edge Detection	0
0	RTSR0	RW	GPIO0 Rising Edge Detection Enable Bit 0: Disable GPIO0 Rising Edge Detection 1: Enable GPIO0 Rising Edge Detection	0

GPIO_IRQPR: Address(0x20)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RO	Not used	0
7	IRQPR7	RO/W1	GPIO7 Interrupt Status Bit 0: No GPIO7 Interrupt Occurred 1: GPIO7 Interrupt Occurred	0
6	IRQPR6	RO/W1	GPIO6 Interrupt Status Bit 0: No GPIO6 Interrupt Occurred 1: GPIO6 Interrupt Occurred	0
5	IRQPR5	RO/W1	GPIO5 Interrupt Status Bit 0: No GPIO5 Interrupt Occurred 1: GPIO5 Interrupt Occurred	0
4	IRQPR4	RO/W1	GPIO4 Interrupt Status Bit 0: No GPIO4 Interrupt Occurred 1: GPIO4 Interrupt Occurred	0
3	IRQPR3	RO/W1	GPIO3 Interrupt Status Bit 0: No GPIO3 Interrupt Occurred 1: GPIO3 Interrupt Occurred	0
2	IRQPR2	RO/W1	GPIO2 Interrupt Status Bit 0: No GPIO2 Interrupt Occurred 1: GPIO2 Interrupt Occurred	0
1	IRQPR1	RO/W1	GPIO1 Interrupt Status Bit 0: No GPIO1 Interrupt Occurred 1: GPIO1 Interrupt Occurred	0
0	IRQPR0	RO/W1	GPIO0 Interrupt Status Bit 0: No GPIO0 Interrupt Occurred 1: GPIO0 Interrupt Occurred	0

GPIOM

OVERVIEW

The GPIOM module has a total of 4 general purpose I/O pins. Support pull up and down selection, data input and output direction control function. Each IO port supports speed selection functions, dual edge interrupt trigger and clear function.

FEATURES

- 4 programmable input/output interfaces.
- Each I/O port supports speed selection functions.

- Each I/O port supports weak pull-up and weak pull-down operations.
- Data input and output direction control.
- Each I/O port supports selection of push-pull or open-drain mode.
- Each I/O port supports dual edge triggering, clearing, and status querying of interrupts.

FUNCTIONAL DESCRIPTION

I/O functions

GPIOM can control the direction of data input and output for each IO by input enable and output enable.

Input function: Configure the corresponding port to GPIOM mode by port reuse, enable the input of the port, and read the level status of each IO port by the GPIOM_IDR register.

Output function: Configure the corresponding port to GPIOM mode by port reuse, enable the output of the port, and configure the output data of each port by the GPIOM_ODR register.

Weak pull-up/down function

Port reuse is used to configure the corresponding port to GPIOM mode. By enabling the weak pull up or weak pull down of the port, users can achieve a weak pull up or weak pull down operation for each IO port. The weak pull up and weak pull down enable control signals of each IO port are independent of each other, and users can configure the weak pull up and weak pull down operation of any IO according to the actual use situation.

I/O function reuse configuration

Configure the value to the register GPIOD_AFR and select the port bit multiplexing mode corresponding to the GPOM. Each IO port has 6 multiplexing modes to choose from. The configuration method for all ports is consistent.

Triggering and clearing interrupts

The port is multiplexed into GPIOM mode to enable gated clock and input enable. The interrupt enable, rising edge enable, or falling edge enable of a port triggers an interrupt when the port produces the expected level change. The interrupt source can be queried by reading the interrupt status register GPIOM_IRQPR. Each interrupt generated by an IO port can be cleared by writing 1 to the bit corresponding to the interrupt status register port separately.

REGISTER LIST

Register Mapping

GPIOM_BASE: 0x40004800

Offset	Register Name	Reset Value	Description
0x00	GPIOM_SPEEDER	0x00	GPIOM output speed control register
0x04	GPIOM_PUPDR	0x0000	GPIOM pull-up/down control register
0x08	GPIOM_IBER	0x00	GPIOM input Enable register
0x0C	GPIOM_OBER	0x00	GPIOM output Enable register
0x10	GPIOM_IDR	xxxx	GPIOM input Data register
0x14	GPIOM_ODR	0x0000	GPIOM output Data register
0x18	GPIOM_AFRL	0x0010_1111	GPIOM AFRL functional reuse register
0x1C	GPIOM_AFRH	0x0	GPIOM AFRH functional reuse register
0x20	GPIOM_IER	0x00	GPIOM interrupt enable register

0x24	GPIOM_FTSR	0x00	GPIOM falling edge detection enable register
0x28	GPIOM_RTSR	0x00	GPIOM rising edge detection enable register
0x2c	GPIOM_PR	0x00	GPIOM interrupt status register

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write, RO/W1: Read only, Write 1 clear

GPIOM_SPEEDER: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RW	Not used	0
8	GPOM8_SPEED	RW	GPIOM8 speed control 0: low 1: fast	0
7	GPOM7_SPEED	RW	GPIOM7 speed control 0: low 1: fast	0
6	GPOM6_SPEED	RW	GPIOM6 speed control 0: low 1: fast	0
5	Reserved	RW	Not used	0
4	GPOM4_SPEED	RW	GPIOM4 speed control 0: low 1: fast	0
3:0	Reserved	RW	Not used	0

GPIOM_PUPDR: Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RW	Not used	0
17:16	PUPDR8	RW	GPIOM8 Pull-Up/Pull-Down Control 00: GPIOM8 No Pull-Up/Pull-Down 01: GPIOM8 Pull-Up Mode 10: GPIOM8 Pull-Down Mode 11: Reserved	0x0
15:14	PUPDR7	RW	GPIOM7 Pull-Up/Pull-Down Control 00: GPIOM7 No Pull-Up/Pull-Down 01: GPIOM7 Pull-Up Mode 10: GPIOM7 Pull-Down Mode 11: Reserved	0x0

13:12	PUPDR6	RW	GPIOM6 Pull-Up/Pull-Down Control 00: GPIOM6 No Pull-Up/Pull-Down 01: GPIOM6 Pull-Up Mode 10: GPIOM6 Pull-Down Mode 11: Reserved	0x0
11:10	Reserved	RW	Not used	0
9:8	PUPDR4	RW	GPIOM4 Pull-Up/Pull-Down Control 00: GPIOM4 No Pull-Up/Pull-Down 01: GPIOM4 Pull-Up Mode 10: GPIOM4 Pull-Down Mode 11: Reserved	0x0
7:0	Reserved	RW	Not used	0

GPIOM_IBER: Address(0x08)

Bit	Symbol	R/W	Description	Default
31:9	Reserved	RW	Not used	0
8	IBER8	RW	GPIOM8 Input Enable Bit 0: Disable GPIOM8 Input 1: Enable GPIOM8 Input	0
7	IBER7	RW	GPIOM7 Input Enable Bit 0: Disable GPIOM7 Input 1: Enable GPIOM7 Input	0
6	IBER6	RW	GPIOM6 Input Enable Bit 0: Disable GPIOM6 Input 1: Enable GPIOM6 Input	0
5	Reserved	RW	Not used	0
4	IBER4	RW	GPIOM4 Input Enable Bit 0: Disable GPIOM4 Input 1: Enable GPIOM4 Input	0
3:0	Reserved	RW	Not used	0

GPIOM_OBER: Address(0x0C)

Bit	Symbol	R/W	Description	Default
31:9	Reserved	RW	Not used	0
8	OBER8	RW	GPIOM8 Output Enable Bit 0: Disable GPIOM8 Output 1: Enable GPIOM8 Output	0
7	OBER7	RW	GPIOM7 Output Enable Bit 0: Disable GPIOM7 Output 1: Enable GPIOM7 Output	0
6	OBER6	RW	GPIOM6 Output Enable Bit 0: Disable GPIOM6 Output	0

			1: Enable GPIOM6 Output	
5	Reserved	RW	Not used	0
4	OBER4	RW	GPIOM4 Output Enable Bit 0: Disable GPIOM4 Output 1: Enable GPIOM4 Output	0
3:0	Reserved	RW	Not used	0

GPIOM_IDR: Address(0x10)

Bit	Symbol	R/W	Description	Default
31:9	Reserved	R	Not used	0
8	IDR8	RO	GPIOM8 Input Data Bit 0: GPIOM8 Input Data Value = 0 1: GPIOM8 Input Data Value = 1	0
7	IDR7	RO	GPIOM7 Input Data Bit 0: GPIOM7 Input Data Value = 0 1: GPIOM7 Input Data Value = 1	0
6	IDR6	RO	GPIOM6 Input Data Bit 0: GPIOM6 Input Data Value = 0 1: GPIOM6 Input Data Value = 1	0
5	Reserved	RW	Not used	0
4	IDR4	RO	GPIOM4 Input Data Bit 0: GPIOM4 Input Data Value = 0 1: GPIOM4 Input Data Value = 1	0
3:0	Reserved	RW	Not used	0

GPIOM_ODR: Address(0x14)

Bit	Symbol	R/W	Description	Default
31:9	Reserved	RW	Not used	0
8	ODR8	RW	GPIOM8 Output Data Bit 0: GPIOM8 Output Data Value = 0 1: GPIOM8 Output Data Value = 1	0
7	ODR7	RW	GPIOM7 Output Data Bit 0: GPIOM7 Output Data Value = 0 1: GPIOM7 Output Data Value = 1	0
6	ODR6	RW	GPIOM6 Output Data Bit 0: GPIOM6 Output Data Value = 0 1: GPIOM6 Output Data Value = 1	0
5	Reserved	RW	Not used	0
4	ODR4	RW	GPIOM4 Output Data Bit 0: GPIOM4 Output Data Value = 0 1: GPIOM4 Output Data Value = 1	0
3:0	Reserved	RW	Not used	0

GPIOM_AFR: Address(0x18)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:28	Reserved	RW	Not used	0x0
27:24	AFR7	RW	GPIOM7 Reuse function Selection 0x6: TIM3_CH3 0x5: UART_TX 0x4: I²C_SDA 0x3: SPI_MOSI 0x0: GPIO	0x0
23:20	AFR6	RW	GPIOM6 Reuse function Selection 0x6: TIM3_CH2 0x5: UART_RX 0x4: I²C_SCL 0x3: SPI_SCK 0x0: GPIO	0x1
19:16	Reserved	RW	Not used	0x0
15:12	AFR4	RW	GPIOM4 Reuse function Selection 0x6: TIM3_CH1 0x3: SPI_MISO 0x0: GPIO	0x1
11:0	Reserved	RW	Not used	0x111

GPIOM_AFR: Address(0x1C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:4	Reserved	RW	Not used	0
3:0	AFR8	RW	GPIOM8 Reuse function Selection 0x6: TIM3_CH4 0x3: SPI_CS 0x0: GPIO	0x0

GPIOM_IER: Address(0x20)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RW	Not used	0
8	IRQENR8	RW	GPIOM8 Interrupt Enable Bit 0: Disable GPIOM8 Interrupt 1: Enable GPIOM8 Interrupt	0
7	IRQENR7	RW	GPIOM7 Interrupt Enable Bit 0: Disable GPIOM7 Interrupt 1: Enable GPIOM7 Interrupt	0

6	IRQENR6	RW	GPIOM6 Interrupt Enable Bit 0: Disable GPIOM6 Interrupt 1: Enable GPIOM6 Interrupt	0
5	Reserved	RW	Not used	0
4	IRQENR4	RW	GPIOM4 Interrupt Enable Bit 0: Disable GPIOM4 Interrupt 1: Enable GPIOM4 Interrupt	0
3:0	Reserved	RW	Not used	0

GPIOM_FTSR: Address(0x24)

Bit	Symbol	R/W	Description	Default
31:9	Reserved	RW	Not used	0
8	FTSR8	RW	GPIOM8 Falling Edge Detection Enable Bit 0: Disable GPIOM8 Falling Edge Detection 1: Enable GPIOM8 Falling Edge Detection	0
7	FTSR7	RW	GPIOM7 Falling Edge Detection Enable Bit 0: Disable GPIOM7 Falling Edge Detection 1: Enable GPIOM7 Falling Edge Detection	0
6	FTSR6	RW	GPIOM6 Falling Edge Detection Enable Bit 0: Disable GPIOM6 Falling Edge Detection 1: Enable GPIOM6 Falling Edge Detection	0
5	Reserved	RW	Not used	0
4	FTSR4	RW	GPIOM4 Falling Edge Detection Enable Bit 0: Disable GPIOM4 Falling Edge Detection 1: Enable GPIOM4 Falling Edge Detection	0
3:0	Reserved	RW	Not used	0

GPIOM_RTSR: Address(0x28)

Bit	Symbol	R/W	Description	Default
31:9	Reserved	RW	Not used	0
8	RTSR8	RW	GPIOM8 Rising Edge Detection Enable Bit 0: Disable GPIOM8 Rising Edge Detection 1: Enable GPIOM8 Rising Edge Detection	0
7	RTSR7	RW	GPIOM7 Rising Edge Detection Enable Bit 0: Disable GPIOM7 Rising Edge Detection 1: Enable GPIOM7 Rising Edge Detection	0
6	RTSR6	RW	GPIOM6 Rising Edge Detection Enable Bit 0: Disable GPIOM6 Rising Edge Detection 1: Enable GPIOM6 Rising Edge Detection	0
5	Reserved	RW	Not used	0
4	RTSR4	RW	GPIOM4 Rising Edge Detection Enable Bit 0: Disable GPIOM4 Rising Edge Detection	0

			1: Enable GPIOM4 Rising Edge Detection	
3:0	Reserved	RW	Not used	0

GPIOM_IRQPR: Address(0x2C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:8	Reserved	RW	Not used	0
8	IRQPR8	RO/W1	GPIOM8 Interrupt Status Bit 0: No GPIOM8 Interrupt Occurred 1: GPIOM8 Interrupt Occurred	0
7	IRQPR7	RO/W1	GPIOM7 Interrupt Status Bit 0: No GPIOM7 Interrupt Occurred 1: GPIOM7 Interrupt Occurred	0
6	IRQPR6	RO/W1	GPIOM6 Interrupt Status Bit 0: No GPIOM6 Interrupt Occurred 1: GPIOM6 Interrupt Occurred	0
5	Reserved	RW	Not used	0
4	IRQPR4	RO/W1	GPIOM4 Interrupt Status Bit 0: No GPIOM4 Interrupt Occurred 1: GPIOM4 Interrupt Occurred	0
3:0	Reserved	RW	Not used	0

AFIO

OVERVIEW

The AFIO module contains the multiplexing of each IO individually as well as the selection of PWM channels.

FEATURES

- Mltiplexing of common ports.
- Selection of PWM channel.

FUNCTIONAL DESCRIPTION

Multiplexing mode switching

- Each IO port supports four modes and the mode switching between IO ports is independent of each other.
- The reuse of its ports is as follows:

Port	MFP=3	MFP=2	MFP=1	MFP=0	default
GPIO0	PWM	/	ANA	GPIO0	MFP=0
GPIO1	PWM	/	ANA	GPIO1	MFP=0
GPIO2	PWM	LINS_TXD	ANA	GPIO2	MFP=0
GPIO3	PWM	LINS_RXD	ANA	GPIO3	MFP=0
GPIO4	PWM	LINM_TXD	ANA	GPIO4	MFP=0

GPIO5	PWM	LINM_RXD	ANA	GPIO5	MFP=0
GPIO6	/	/	/	GPIO6	MFP=0
GPIO7	/	/	/	GPIO7	MFP=0

Selection of PWM

Each port of AW32F020 can be multiplexed into PWM mode. When the port is multiplexed into PWM mode, each GPIO port can select the PWM channel to be output by the AFIO.AFIO_PWMSEL register, so that all three PWMS can be output from any IO port.

Take GPIO1 as an example, configure MFP register to 0x30 to select PWM output mode. If PWMSEL is chosen to be 0x0, then PWM0 is output from GPIO1 port; If 0x4, PWM1 output from GPIO1 port; If 0x8, PWM2 outputs from GPIO1 port. This is the same for all IO port configurations.

REGISTER LIST

Register Mapping

Offset	Register Name	Description	Reset Value
0x00	AFIO_MFPR	GPIO Alternate Function Register	0x00
0x04	AFIO_PWMSELR	PWM Selection Register	0x00

Register Detailed Description

AFIO_BASE : 0x50000800

RO: Read only, W: Write only, RW: Read/Write.

AFIO_MFPR: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:28	GPIO7_AFR	RW	GPIO5 Alternate Function Bit	0x0
27:24	GPIO6_AFR	RW	GPIO5 Alternate Function Bit	0x0
23:20	GPIO5_AFR	RW	GPIO5 Alternate Function Bit	0x0
19:16	GPIO4_AFR	RW	GPIO4 Alternate Function Bit	0x0
15:12	GPIO3_AFR	RW	GPIO3 Alternate Function Bit	0x0
11:8	GPIO2_AFR	RW	GPIO2 Alternate Function Bit	0x0
7:4	GPIO1_AFR	RW	GPIO1 Alternate Function Bit	0x0
3:0	GPIO0_AFR	RW	GPIO0 Alternate Function Bit	0x0

AFIO_PWMSELR: Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:12	Reserved	RO	Not used	0
11:10	GPIO5_PWM_SEL	RW	GPIO5 PWM Channel Selection 00: PWM channel0 01: PWM channel1 10: PWM channel2 11: PWM channel3	0x0

9:8	GPIO4_PWM_SEL	RW	GPIO4 PWM Channel Selection 00: PWM channel0 01: PWM channel1 10: PWM channel2 11: PWM channel3	0x0
7:6	GPIO3_PWM_SEL	RW	GPIO3 PWM Channel Selection 00: PWM channel0 01: PWM channel1 10: PWM channel2 11: PWM channel3	0x0
5:4	GPIO2_PWM_SEL	RW	GPIO2 PWM Channel Selection 00: PWM channel0 01: PWM channel1 10: PWM channel2 11: PWM channel3	0x0
3:2	GPIO1_PWM_SEL	RW	GPIO1 PWM Channel Selection 00: PWM channel0 01: PWM channel1 10: PWM channel2 11: PWM channel3	0x0
1:0	GPIO0_PWM_SEL	RW	GPIO0 PWM Channel Selection 00: PWM channel0 01: PWM channel1 10: PWM channel2 11: PWM channel3	0x0

LOW-SIDE SWITCH

FEATURES

- 4 channels high voltage low-side switch, up to 60mA maximum output current per channel
- Each channel could be configured as high voltage open-drain output
- Each channel could be configured as ADC sampling port
- Each channel independent 9bit current control, STEP=120μA
- The resolution of PWM can be adjusted up to 16bit
- Support 4-channel PWM modulation
- The PWM frequency ranges from 80 Hz to 250Hz
- The PWM clock is derived from the system clock and supports 1/2/4/8/16/64/256/1024 frequency division
- Support LS voltage detection, LS voltage drop detection 0 V~8 V configurable
- The configuration characteristics of the rise threshold PRISE and fall threshold PFALL are as follows.
  - PRISE<PFALL, corresponding to PWM output duty cycle PWM_DUTY=(PFALL- PRISE)/PRIOD
  - PRISE=0 and PFALL=PERIOD, corresponding to 100% duty cycle of PWM output
  - PRISE≥PFALL, which corresponds to 0% duty cycle of PWM output

FUNCTIONAL DESCRIPTION

Current sink configuration

AW32F020 has 4 independent high-precision current sink modules, and the current variation range is 0~60mA

with 120μA step. The reference current of the module is provided by LS IBIAS. The LS IBIAS module is calibrated at the factory, and the calibration value is stored in the built-in FLASH, which is loaded into the PMUIBIAS_TRIM register (which is located in the SYSCTRL module) after the program is started. The maximum current of LSx is calculated by:

$$I_{LSx}\ (max) = LSBRx\_TRIM * 120uA$$

Where x=0,1,2,3, and LSBRx_TRIM is the register configuration entry (located in the SYSCTRL module).

The current of LSx is calculated by:

$$I_{LSx} = I_{LSx}(max) * PWM\_DUTY = LSBRx\_TRIM * 120uA * PWM\_DUTY$$

Where PWM_DUTY is the duty cycle of the input PWM wave, see PWM section for details.

The LS module configuration steps are as follows:

1) BUFEN set and wait for 50μs (BUFEN is located in the ACR register of the PMU module);
2) IBIASEN is set to enable LS IBIAS module (IBIASEN is located in the ACR register of PMU module);
3) Set EN_LSx_BIAS to enable LSx BIAS. Configure LSBRx_TRIM to control the maximum current of LSx.
4) PWM related registers are configured and LS_EN is set to generate PWM waveform and control the current of LSx.

PWM control

The counting clock of the PWM is obtained by the system clock frequency division. User can configure the PWMFRQ register to flexibly select the clock frequency.

After the LSBRx_TRIM configuration is completed, the current of each LS can be controlled by adjusting the PWM duty cycle. The PWM duty cycle depends on the configuration of three registers: PERIOD, PFALL and PRIZE, which can be calculated by:

$$PWM\_DUTY = \frac{PFALL - PRISE}{PERIOD}$$

Where the value of PERIOD should be greater than 2, and the value of PFALL should be no less than the value of PRISE, otherwise the waveform of PWM may not meet the expectation.

The frequency division coefficient, count period, rise threshold, fall threshold, and reverse phase configuration terms are allowed to be modified during operation. In order to enable multi-channel LS parameters to be configured simultaneously and prevent PWM waveform mutation, these values are updated at the end of the current count. By continuously configuring the registers and adjusting the waveform of the PWM, user controls the current of LSx and obtains different LS effects.

ADC sample

LS0~3 could be configured as ADC sample port. The configuration steps of LS voltage detection are as follows:

1) BUFEN set and wait for 50μs (BUFEN is located in the ACR register of the PMU module);
2) IBIASEN is set to enable LED IBIAS module (IBIASEN is located in the ACR register of PMU module);
3) Set EN_VFW_BIAS to enable VFW BIAS module.
4) Select the appropriate GAIN value, and then set EN_VFW_DECT to enable the detection module.

REGISTER DETAILED DESCRIPTION

LS_BASE : 0x50000400

RO: Read only, W: Write only, RW: Read/Write.

PWM Control register (PWM_CR0): Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	Reserved	RO	Not used	0
15:14	PWMFORCE3	RW	PWM3 force enable	0

			bit[1]: PWM3 enable 0: Diasble 1: Enable bit[0]: PWM3 output 0: LOW 1: HIGH	
13:12	PWMFORCE2	RW	PWM2 force enable bit[1]: PWM2 enable 0: Diasble 1: Enable bit[0]: PWM2 output 0: LOW 1: HIGH	0
11:10	PWMFORCE1	RW	PWM2 force enable bit[1]: PWM1 enable 0: Diasble 1: Enable bit[0]: PWM1 output 0: LOW 1: HIGH	0
9:8	PWMFORCE0	RW	PWM0 force enable bit[1]: PWM0 enable 0: Diasble 1: Enable bit[0]: PWM0 output 0: LOW 1: HIGH	0
7	EN_LS	RW	LS enable 0: Diasble 1: Enable	0
6:3	INVERT	RW	PWM invert INVERT[0]: PWM_OUT0 invert enable 0: Diasble 1: Enable INVERT[1]: PWM_OUT1 invert enable 0: Diasble 1: Enable INVERT[2]: PWM_OUT2 invert enable 0: Diasble 1: Enable INVERT[3]: PWM_OUT3 invert enable 0: Diasble 1: Enable	0
2:0	PWMFRQ	RW	PWM Clock frequency 000: DIV1 001: DIV2 010: DIV4 011: DIV8	0

			100: DIV16 101: DIV64 110: DIV256 111: DIV1024	

PWM Control register (PWM_CR1): Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	Reserved	RO	Not used	0
15:0	PERIOD	RW	Maximum count value of PWM	0

PWM Update register (PWM_PUR): Address(0x08)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:1	Reserved	RO	Not used	0
0	UPDATE	RW	LS parameter update enable 0: Diasble 1: Enable	0

PWM Count register (PWM_PCR): Address(0x0C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	Reserved	RO	Not used	0
15:0	PWMCNT	RO	PWM count value	0

Threshold control register(PWM_THR0): Address(0x10)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	PRISE0	RW	LS0 rise threshold selection	0
15:0	PFALL0	RW	LS0 descent threshold selection	0

Threshold control register(PWM_THR1): Address(0x14)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	PRISE1	RW	LS1 rise threshold selection	0
15:0	PFALL1	RW	LS1 descent threshold selection	0

Threshold control register(PWM_THR2): Address(0x18)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	PRISE2	RW	LS2 rise threshold selection	0
15:0	PFALL2	RW	LS2 descent threshold selection	0

Threshold control register(PWM_THR3): Address(0x1C)

Bit	Symbol	R/W	Description	Default
31:16	PRISE3	RW	LS3 rise threshold selection	0
15:0	PFALL3	RW	LS3 descent threshold selection	0

LS control register(LS_LCR): Address(0x20)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:7	Reserved	RO	Not used	0
6	EN_LS3_BIAS	RW	LS3 bias enable 0: Diasble 1: Enable	0
5	EN_LS2_BIAS	RW	LS2 bias enable 0: Diasble 1: Enable	0
4	EN_LS1_BIAS	RW	LS1 bias enable 0: Diasble 1: Enable	0
3	EN_LS0_BIAS	RW	LS0 bias enable 0: Diasble 1: Enable	0
2:1	SRF	RW	Drop rate adjustment of LS DRIVER Note: 00 is the fastest speed	0
0	SRR	RW	LS DRIVER rise rate adjustment Note: 0 is the fastest rate	0

VFW control register(VFW_VCR0): Address(0x24)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:3	Reserved	RO	Not used	0
2	EN_VFW_DECT	RW	VFW current sensing enable 0: Diasble 1: Enable	0
1	EN_VFW_BIAS	RW	VFW bias enabled 0: Diasble 1: Enable	0
0	GAIN	RW	VFW gain selection enabled 0: 1/4 1: 1/8	0

VFW control register(VFW_VCR1): Address(0x28)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:2	Reserved	RO	Not used	0
1	VFW_PWM_FPLS_EN	RW	At PWM low level, the VFW test current is	1

			enabled 0: Diasble 1: Enable	
0	VFW_PWM_RPLS_EN	RW	At PWM high level, VFW test current enabled 0: Diasble 1: Enable	0

VFW test register(VFW_VTR0): Address(0x2C)

Bit	Symbol	R/W	Description	Default
31:0	VFW_TEST_EN	RW	Write 32'h33445566 into test mode, other values exit test mode, VFW test mode state 0: Non-VFW test mode 1: VFW test mode	0

VFW test register(VFW_VTR1): Address(0x30)

Bit	Symbol	R/W	Description	Default
31:4	Reserved	RO	Not used	0
3	VFW_TEST_CURRENT_EN	RW	In VFW test mode, the VFW current is enabled 0: Disable the enable 1: Turn on the enable	1
2	VFW_TEST_DECT_PULSE_N	RW	In VFW test mode, VFW detection is enabled 0: Turn on the enable 1: Disable the enable	0
1:0	VFW_TEST_SEL_LS	RW	When VFW_TEST_EN is active, the VFW current is selected to enable the LS channel 00: Select LS0 01: Select LS1 10: Select LS2 11: Select LS3	0

ADC

FEATURES

- 12bit precision analog ADC
- Single input
- Analog channel support
  - 1/32 VBAT
  - GPIO0~5
  - LS0~3
  - VDD1V5
  - VREF

- Temperature sensor

- Digital channel supports analog channel free mapping, and supports up to 4 digital channels continuous scanning
- Support internal interface (PWM) trigger scan action, and software trigger scan action
- Support single-channel interrupt and multi-channel interrupt
- Support multiple scan modes, including single scan, continuous scan, cycle scan

FUNCTIONAL DESCRIPTION

Source of trigger

- Software trigger
- The falling edge, rising edge or double edge of PWMx (0~2) can be selected as the trigger source

Scan mode

Single scan mode: Trigger ADC scan SEQ_LEN ADC channels. After the scanning action is completed, the ADC channel will not be scanned even if the trigger signal is generated again. The following registers are configured to enter single-scan mode: TRIG_MODE in ADC_CR is set to 1, and TRIG_VALID in ADC_CR is set to 1.

Continuous scan mode: Trigger ADC scan SEQ_LEN ADC channels. The SEQ_LEN ADC channels will rescan if the trigger signal is generated again. The following registers are configured to enter continuous scan mode: TRIG_MODE in ADC_CR is set to 0, and TRIG_VALID in ADC_CR is set to 1.

Cyclic scan mode: Trigger ADC scan SEQ_LEN ADC channels. After scanning, SEQ_LEN ADC channels are rescanned. The following registers are configured to enter the cyclic scan mode: CONT is set to 1 in ADC_CR and TRIG_VALID is set to 1 in ADC_CR.

Sampling time configuration

To collect different analog signals, different sampling times need to be configured, as shown in the following table.

Analog signal	Sampling time/μs
Chip voltage	8μs
Chip temperature	8μs
GPIO2~5	8μs
LS0~3	32μs
VDD1P5	8μs
VREF	8μs

When the system clock is configured to 16M and CH0 takes the LS voltage, STUP_DLY should be configured to 0xFF and SAMPLE to 0xFF. When the LS voltage is collected without CH0, it is necessary to configure CHANNEL_DLY as 0xFF and SAMPLE as 0xFF.

Analog quantity calculation

1) Power supply voltage calculation

$$\mathrm{VBAT} = \frac{\mathrm{code} * \mathrm{VREF} * 32}{4096}$$

Where VREF is selected by VREF_SEL in ADC_CR.

2) Junction voltage calculation of LS

$$\mathrm{VFW} = \frac{\mathrm{code} * \mathrm{VREF} * \mathrm{Gain}}{4096}$$

Where VREF is selected by VREF_SEL in ADC_CR, Gain is selected by VFW_VCR0_b.GAIN in LS_CTRL, if the LS junction voltage is 0~4.8V, GAIN is selected by 4, if the LS junction voltage is 0~9.6V, GAIN is selected

by 8.

3) Chip temperature calculation

$$\text{Vtemp} = \frac{\text{code} * \text{VREF}}{4096}$$

VREF is selected by VREF_SEL in ADC_CR, and the coefficient of Vtemp converted to temperature is 0.44643°C/mV. For example, if the voltage collected to the temperature sensor at room temperature 25°C is Vtemp_type, the chip temperature Tchip under the temperature sensor voltage VtempT is:

$$\text{Tchip} = 0.44643 * (\text{VtempT} - \text{Vtemp\_type}) + 25$$

4) Other

$$\text{V} = \frac{\text{code} * \text{VREF}}{4096}$$

VREF is selected by VREF_SEL in ADC_CR.

Operating mode

The operating mode of the ADC is shown in the following figure. The PWM falling edge is selected as the trigger source of the ADC operation in the register setting. The number of scanned channels is chosen to be 3. When the PWM falling edge comes, the ADC will scan CH0/1/2 in turn, and the analog channels mapped by CHx can be freely combined, for example CH0 selects 1/32 VBAT, CH1 selects LS1, and CH2 selects temperature sensor.

After the PWM falling edge comes, after a TTRIG_DLY+TSTUP_DLY delay, the CH0 channel starts scanning. The setup time of scanning is controlled by the CHx_SMPL register, and the specific time is configured according to the sampling time. After the establishment time of CHx_SMPL, the ADC starts to convert the data, which requires 13 ADC_CLK cycles, in which the ADC_CLK cycle is controlled by the CONV_CLK_DIV frequency division coefficient. After the conversion is complete, TCHANNEL_DLY starts the next channel conversion.

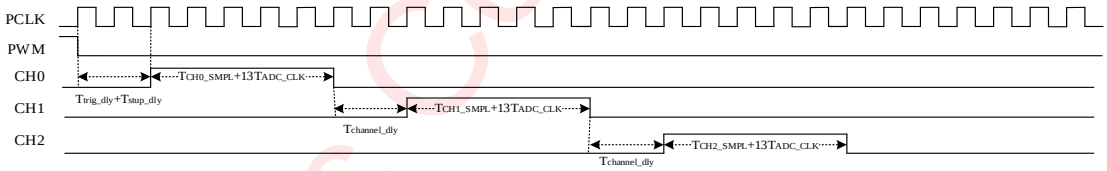


Figure 9 An example diagram of the ADC operating mode

REGISTER LIST

Register Detailed Description

ADC_CTRL_BASE : 0x50000700

RO: Read only, W: Write only, RW: Read/Write.

ADC_SR: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:3	Reserved	RO	Not used	0
2	START	RW	ADC start working status 0: The ADC can be started 1: The ADC is working	0
1	EOSEQ	RW	ADC end of sequence scan process 0: Scan in progress 1: The scan is complete	0
0	EOC	RW	ADC end of channel scan process	0

			0: Scan in progress 1: The scan is complete	

ADC_CR: Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:22	Reserved	RO	Not used	0
21:20	VREF_SEL	RW	ADC reference voltage select 0x0: Select VBG as VREF 0x1: Select 2*VBG as VREF 0x2: Select VDD 3.3V as VREF	0x1
19	CONT	RW	ADC continuous scan mode 0: adc stop when sequence scan is over 1: adc will start from first channel again when sequence is over	0
18:16	CONV_CLK_DIV	RW	clock divider of sysclk for analog-ADC 000: div by 4 011: div by 8 100: div by 16 101: div by 32 110: div by 64 111: div by 128	0x0
15	TRIG_MODE	RW	ADC trigger valid once select 0: adc trigger will always be valid 1: adc trigger will be invalid when conversion is done	0
14	TRIG_VALID	RW	ADC trigger valid enable bit 0: adc trigger is invalid 1: adc trigger is valid	1
13:12	PWM_SEL	RW	ADC PWM trigger select 00: trigger by PWM0 01: trigger by PWM1 10: trigger by PWM2 11: trigger by PWM3	0x0
11:9	EXT_SEL	RW	ADC trigger select 000: trigger by posedge of pwm signal 001: trigger by negedge of pwm signal 010: trigger by both edge of pwm signal 111: trigger by software	0x7
8	SW_TRIG	RW	ADC software trigger When EXT_SEL[2:0] is 3'b111 0: The software trigger signal cannot trigger the ADC scan 1: The software trigger signal can trigger the ADC scan	0
7:2	Reserved	RO	Not used	0
1	EOSEQIE	RW	ADC end of sequence interrupt enable 0: Disable 1: Enable	0

0	EOCIE	RW	ADC end of channel interrupt enable 0: Disable 1: Enable	0

ADC_DLYR: Address(0x08)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:24	Reserved	RO	Not used	0
23:16	TRIG_DLY	RW	ADC start delay when trigger happens delay time = Tfclk*(TRIG_DELAY+2)	0x10
15:8	CHANNEL_DLY	RW	ADC channel delay when scan channel switch form one to another delay time = Tfclk*(CHANNEL_DLY+4)	0x00
7:0	STUP_DLY	RW	ADC start up delay time before sample phase delay time = Tfclk*(STUP_DLY+2)	0x10

ADC_SMPR: Address(0x0C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:24	CH3_SMPL	RW	ADC channel 3 sampling time configuration sample time = Tfclk*(CHx_SMPL+1)	0x0
23:16	CH2_SMPL	RW	ADC channel 2 sampling time configuration sample time = Tfclk*(CHx_SMPL+1)	0x0
15:8	CH1_SMPL	RW	ADC channel 1 sampling time configuration sample time = Tfclk*(CHx_SMPL+1)	0x0
7:0	CH0_SMPL	RW	ADC channel 0 sampling time configuration sample time = Tfclk*(CHx_SMPL+1)	0x0

ADC_SQR1: Address(0x10)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:2	Reserved	RO	Not used	0
1:0	SEQ_LEN	RW	ADC scan sequence length 00: ONE_CHANNEL 01: TWO_CHANNELS 10: THREE_CHANNELS 11: FOUR_CHANNELS	0x0

ADC_SQR2: Address(0x14)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	Reserved	RO	Not used	0
15:12	CH3_SEL	RW	ADC scan channel selection, refer to CH0_SEL	0x0
11:8	CH2_SEL	RW	ADC scan channel selection, refer to CH0_SEL	0x0
7:4	CH1_SEL	RW	ADC scan channel selection, refer to CH0_SEL	0x0

3:0	CH0_SEL	RW	ADC scan channel select 0000: Reserved 0001: 1/32*VBAT 0010: temp sensor 0011: GPIO0 0100: GPIO1 0101: GPIO2 0110: GPIO3 0111: GPIO4 1000: GPIO5 1001: VREF 1010: VDD1V5 1011: VDD3V3 1100: VFW_LS0 1101: VFW_LS1 1110: VFW_LS2 1111: VFW_LS3	0x0

ADC_DR1: Address(0x18)

Bit	Symbol	R/W	Description	Default
31:28	Reserved	RO	Not used	0
27:16	ADC_DATA1	RO	ADC channel 1 conversion data	0x0
15:12	Reserved	RO	Not Used	0
11:0	ADC_DATA0	RO	ADC channel 0 conversion data	0x0

ADC_DR2: Address(0x1C)

Bit	Symbol	R/W	Description	Default
31:28	Reserved	RO	Not used	0
27:16	ADC_DATA3	RO	ADC channel 3 conversion data	0x0
15:12	Reserved	RO	Not Used	0
11:0	ADC_DATA2	RO	ADC channel 2 conversion data	0x0

LIN BUS

LIN bus integrates LIN_PHY and LIN_CONTROLLER to realize LIN communication. x standard, adaptive baud rate, 1-20KB/s transceiver rate, maximum support 115200b/s one-way receiving rate. Supports LIN automatic addressing.

FEATURES

- Compatible with LIN2.x and SAE J2602 standards
- Support for automatic addressing
- Support adaptive baud rate, rate range 1~20kb/s, maximum support 115200b/s receiving rate
- Support wake-up detection in low power mode
- Multiple timeout detection mechanism and hardware circuit protection function

- Supports multiple frame format error detection mechanisms
- Configurable protocol frame length

FUNCTIONAL DESCRIPTION

LIN_PHY

LIN_PHY integrates RX module and TX module to convert high voltage signal of LIN port into low voltage digital signal to LIN controller. The delay and other related timing requirements of the digital signal output by the LIN_PHY meet the ISO17987 specification.

RXD filter

In order to prevent the electromagnetic pulse interference to LIN bus in industrial environment or other application environment, DIG_PHY integrates a digital filter driven by the system clock.

The high level filter and low level filter support free configuration, which facilitates the user's flexible choice in different application environments.

Baud rate

The slave supports adaptive baud rate function, and users can receive protocol frames from the host without configuring baud rate in advance. Depending on the system clock frequency, the slave can maximally support 115200b/s one-way reception or 1-20kb /s two-way transmission.

Frame length and validation

The slave supports protocol frames with a data length of 0 to 8 bytes configured by the LIN_CR1.DATA_LENGTH register. But when the register is configured as 4'b1111, the data segment length of the protocol frame is determined by Bit5 and Bit4 of the received ID field.

ID[5]	ID[4]	Data length
0	0	2
0	1	2
1	0	4
1	1	8

The slave supports both classical and enhanced verification modes.

Low power mode

The slave supports low power mode. After LIN bus timeout, LIN controller will automatically enter the low power mode, LIN_SR.SLEEP is set, and an interrupt notification is sent. The user can configure the chip to enter the low power mode, turn off the analog module or unnecessary digital module, so as to save system power consumption.

After entering the low-power mode, the LIN controller uses the low-speed clock to detect whether there is a valid wakeup instruction on the LIN bus. After detecting the wake-up command, the chip automatically enters the wake-up process and turns on the high-speed clock and other simulator components.

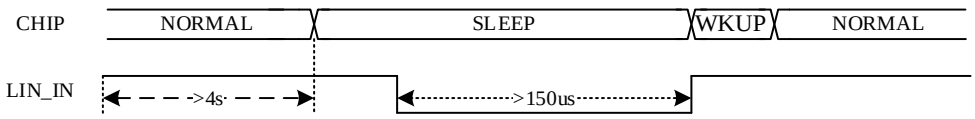


FIG. 9 Timing diagram of low power consumption of LIN wake-up chip

The wakeup command refers to a dominant level that exceeds the set width of the LIN_TSR.WKUP_REC_THRES register and a valid rising edge to ensure that the wakeup command was issued by the device and not due to a bus grounding error.

Timeout detection

Multiple timeout detection counters are integrated in the LIN controller to detect the timeout of the bus or internal signal, mainly including:

- LIN bus timeout:

The LIN bus is inactive and the explicit or invisible level is held for more than a threshold time (configured by the register LIN_TSR.BUS_TIMEOUT_DET_THRES). The LIN controller detects this timeout state and generates an interrupt. At the same time, the LIN controller monitors the wake-up signal of the LIN bus.

- LIN bus explicit timeout:

The LIN bus is inactive and the dominance level is held for more than a threshold time (configured by the register LIN_TSR.BUS_TIMEOUT_DET_THRES). This means that the LIN bus is very likely to be in the ground state; When the LIN_PHY_CR.PU30K_AUTOOFF_EN register is 1, if the LIN bus is dominant timeout, it will automatically disconnect the connection between the pull-up resistor and the LIN bus to prevent the overshoot of large current from causing damage to the device.

- Wake-up response timeout:

When the LIN host is sleeping, the LIN controller can control the bus to initiate a wake-up command transmission actively. At the same time, after the wake-up command transmission is completed, the LIN controller starts to detect whether the host is successfully awakened. If the wake-up is successful, the host will initiate a protocol frame transmission. If the controller does not detect a valid protocol frame within a set threshold time (configured by LIN_TSR.WKUP_REPEAT_THRES), wake-up has failed, and an interrupt is generated so that the user can decide whether to wake up again.

- TXD dominant timeout

The TXD data of analog transceiver comes from the internal logic of LIN controller. When LIN_TXD is 0, LIN_IN is 0, and there is valid data transmission on the bus. If LIN_TXD is 0 more than 64ms, it is considered that there is an error in the internal circuit of the controller, at this time, there will be an interrupt, and it is up to the user to decide whether to reset the controller. The transmitter circuit of the analog transceiver will automatically disconnect in the case of TXD timeout to prevent the damage of the device by current overshoot.

Error detection

The LIN controller can detect a variety of errors within the protocol frame or internal logic, including:

- Controller response timeout error

When the controller receives the Header field sent by the host, it first determines the rate of the current frame through the Break-Sync field, and then receives the PID field according to the determined rate and checks whether the checksum in the PID field is wrong. If the receiver is correct, it generates an interrupt to notify the user that the correct frame header has been received. It requires the user to indicate whether the current frame is received or sent, the length and content of the frame data field, and the validation method. Once the user has determined what to receive or send, the LIN_CR1.DATA_ACK bit is written to inform the controller that the current frame configuration is complete.

Due to the maximum length limit of LIN protocol, after the controller receives the frame header and generates an interrupt, if the host has begun to send data and the first byte of data has been sent before the DATA_ACK configuration, the user has not written DATA_ACK at this time, it is considered that the controller response time is out and the current frame cannot be correctly parsed.

- Frame response timeout error

When the controller receives the Header field from the host, it first determines the current frame rate using the Break-Sync field, and then prepares to receive the PID field and the data field. If the PID field or data field is not received within the maximum length specified by the protocol, a frame response timeout error will be generated.

- Sending and receiving path error

The controller will continuously compare the values of RXD and TXD when sending the protocol frame. If they are not equal at a certain time, it is considered that there is more than one node driving the LIN bus at present,

resulting in a sending and receiving path error.

- Frame missing error

When the controller receives the Header field from the host, it first determines the current frame rate using the Break-Sync field, and then prepares to receive the PID field and the data field. If the PID field or data field is not received within the maximum length specified in the protocol, a frame missing error will occur.

- Start/stop bit error
- Testing and error

Automatic addressing

The chip integrates LIN_SW and two LIN buses: LIN_IN and LIN_OUT. LIN_SW is controlled by the register LIN_PHY_CR.AUTO_ADDRESS_EN to link LIN_IN and LIN_OUT together inside the chip for automatic addressing purposes.

CONFIGURATION FLOW

LIN Slave initializes the configuration process

1) Start the LIN module clock

The turning on and off of the LIN module clock can be configured by means of the RCC.RCC_PENR register.

2) Enable the data sending and receiving function of LIN module

Configure LIN ANALOG_PHY to enable the data sending and receiving functions of LIN Master module and LIN Slave module.

3) Configure LIN interrupts

The interrupt for the LIN Slave module can be configured via the LINS.LIN_IER register.

4) Enable the 30kΩ pull-up resistor of the LIN Slave module

The 30kΩ pull-up resistor of the LIN Slave module can be configured via the LINS.LIN_PHY_ANR register.

5) Enable automatic addressing function

The automatic addressing of the LIN module can be turned on or off by means of the LINS.LIN_PHY_CR register.

6) Configure the LIN bus timeout threshold

The LIN Slave module bus timeout threshold can be configured via the LINS.LIN_TSR register.

7) Configure the wake-up signal retransmission interval

The interval at which the wake-up signal is resent can be configured by means of the LINS.LIN_TSR register.

8) Configure the LIN Slave to send data monitoring function

The send data monitoring function of the LIN Slave module can be configured through the LINS.LIN_CR2 register.

9) Configure the communication baud rate (adaptive baud rate or adaptive Baud rate)

The LIN Slave module adaptive baud rate function can be configured via the LINS.LIN_CR2 register.

When the adaptive baud rate function of the LIN Slave module is disabled, the fixed baud rate can be configured through the LINS.LIN_RTR register.

LIN Slave data receiving process

The LIN Slave data receiving process is as follows:

1) Generate an interrupt request

Upon receiving the correct ID field, the LIN slave generates an interrupt request to notify the program that a new instruction has been received.

2) Configure the frame orientation

The ID is read, and the instruction is parsed to identify the type of operation for the current frame. When the

operation type of the current frame is recognized as receive, the Bit5 of the LINS.LIN_CR1 register needs to be configured to 0 and the frame direction is configured to receive.

3) Configure the length of received data

The received data length is configured in the LINS.LIN_CR1 register.

4) Configure the checksum type of the received data

The type of the received data checksum can be configured via the LINS.LIN_CR1 register.

5) Frame data confirmation

After the frame orientation, the length of the received data, and the checksum type of the received data are configured, the Bit4 of the LINS.LINS_CR1 register needs to be configured to 1 to inform the controller to continue parsing the frame content.

6) Wait for data reception to complete

After the hardware receives the data, it stores the data in the buffer and completes the verification. If the verification data is correct, the receiving data completion interrupt will be generated.

7) Parsing data at the application layer

The application layer reads the received data from the LINS.LIN_DR1 and LINS.LIN_DR2 registers and parses the received data.

LIN Slave data sending process

The LIN Slave data sending process is as follows:

1) Generate an interrupt request

Upon receiving the correct ID field, the LIN slave generates an interrupt request to notify the program that a new instruction has been received.

2) Configure the frame orientation

The ID is read, and the instruction is parsed to identify the type of operation for the current frame. When the operation type of the current frame is recognized as send, the Bit5 of the LINS.LIN_CR1 register needs to be configured to 1 and the frame direction is configured to send.

3) Configure the length of data to be sent

The length of the sent data is configured in the LINS.LIN_CR1 register.

4) Configure the checksum type of the sent data

The type of the received data checksum can be configured via the LINS.LIN_CR1 register.

5) Configure to send data

Write the data to be sent to the LINS.LIN_DR1 and LINS.LIN_DR2 registers.

6) Frame data confirmation

After the frame direction, the length of the data to be sent, the checksum type of the data to be sent, and the data to be sent are configured, the Bit4 of the LINS.LINS_CR1 register needs to be configured to 1 to notify the controller to start the data sending process.

7) Wait for the data to be sent

After the completion of sending data, the sending data completion interrupt will be generated.

LIN Slave sleep and wake up process

When the LIN automatic sleep function is enabled, when there is no activity on the LIN bus and the duration of the dominant or invisible level exceeds the set automatic sleep time, the controller will automatically enter the sleep state.

The LINS.LIN_CR2 register is configured to enable and disable the LIN automatic sleep function.

The duration of the stealth level on the LIN bus can be configured via the LINS.LIN_CR1 register.

When receiving a sleep command sent by the host, the Bit6 of the LINS.LIN_CR1 register can also be configured to 1 to put the controller to sleep.

When the LIN controller is in the sleep state and detects a valid wake-up signal, it will generate a LIN wake-up interrupt. When the LIN wake-up interrupt is generated, the Bit6 of the LINS.LIN_CR1 register needs to be configured to 0 to make the controller exit the sleep state.

REGISTER DETAILED DESCRIPTION

LIN_S_BASE : 0x50000500

RO: Read only, W: Write only, RW: Read/Write.

LIN_DR1: Address(0x00)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:24	DATA4	RW	Data Buffer 4. 4th byte of the reponse data	0x0
23:16	DATA3	RW	Data Buffer 3. 3rd byte of the reponse data	0x0
15:8	DATA2	RW	Data Buffer 2. 2nd byte of the reponse data	0x0
7:0	DATA1	RW	Data Buffer 1. 1st byte of the reponse data	0x0

LIN_DR2: Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:24	DATA8	RW	Data Buffer 8. 8th byte of the reponse data	0x0
23:16	DATA7	RW	Data Buffer 7. 7th byte of the reponse data	0x0
15:8	DATA6	RW	Data Buffer 6. 6th byte of the reponse data	0x0
7:0	DATA5	RW	Data Buffer 5. 5th byte of the reponse data	0x0

LIN_IDR: Address(0x08)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:6	Reserved	RO	Not used	0
5:0	PID	RW	PID Buffer	0x0

LIN_SR: Address(0x0C)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:30	Reserved	RO	Not used	0
29	BUS_TIMEOUT_INT_STATUS	RO	Bus timeout interrupt status.	0
28	TXD_TIMEOUT_INT_STATUS	RO	Txd timeout interrupt status.	0
27:25	Reserved	RO	Not used	0

24	WKUP_INT_STATUS	RO	Wake up receive status.	0
23:20	Reserved	RO	Not used	0
19	TXD_TIMEOUT	RO	Txd timeout status. This bit is set to 1 if the TXD is continuously 0 for more than 64ms, and clear to 0 when the TXD is 1. TXD of LIN_PHY is turned off after TXD timeout	0
18	WKUP_TIMEOUT	RO	Wake up timeout status. This bit is set to 1 and written 1 clear 0 if the frame header is not received within a certain period of time after the controller sends the wake-up signal	0
17	ACK_TIMEOUT	RO	Data Acknowledge timeout status. The frame data confirms the status This bit is set when the user does not acknowledge the frame data after the frame data request is valid and before the first byte of the frame data is received. Write 1 clear.	0
16	RESPONSE_TIMEOUT	RO	None response timeout status. This bit is set when the PID or data area is missing due to the frame. Write 1 clear.	0
15	TXD_ERR_DATA	RO	TXD Compare Error In Data/Start Bit. This bit is set when the controller sends data or when there is an error that TXD is not the same as RXD. Write 1 clear.	0
14	TXD_ERR_STOP	RO	TXD Compare Error In Stop Bit. This bit is set when there is an error that TXD is not the same as RXD when the controller sends the stop bit. Write 1 clear.	0
13	TXD_ERR	RO	Txd Compare Error. This bit is set when LIN_TXD is detected to be different from LIN_RXD, and a 1 is written to clear a 0. This bit indicates that the LIN bus has multiple nodes sending data at the same time.	0
12	FRAME_ERR	RO	Frame Byte Filed Error. This bit is set when the data area is too short or the data area is too long to cause a timeout. Write 1 clear. Note: Too long refers to more than 127 Tbit.	0
11	STARTB_ERR	RO	Start Bit Filed Error. This bit is set if the start bit width is detected to be less than 1Tbit. Write 1 clear.	0
10	PID_ERR	RO	Pid Parity Check Error. This bit is set in the case of PID data	0

			verification error. Write 1 clear.	
9	CHECK_ERR	RO	CheckSum Error. This bit is set in the case of frame checksum failure. Write 1 clear.	0
8	STOPB_ERR	RO	Stop Bit Error. This bit is set if the stop bit is detected to be 0. Write 1 clear.	0
7	BUS_ACTIVE	RO	LIN Bus Active Status. This bit is set when a valid frame header is received and reset when the frame is over. This bit can also be cleared by writing the LIN_CR1->PROC_RESET bit	0
6	BUS_TIMEOUT	RO	LIN Bus timeout Status. This bit is set when the LIN bus has been inactive for a certain period of time (LIN_TSR->BUS_TIMEOUT_DET_THRES) and the LIN controller automatically goes to sleep after this bit	0
5	BUS_DOMINANT_TIMEOUT	RO	LIN Bus Dominant timeout Status. This bit is set after the LIN bus has maintained the dominant level for a certain period of time (LIN_TSR->BUS_TIMEOUT_DET_THRES), and the LIN controller will automatically go to sleep after this position	0
4	ABORTED	RO	LIN Core Aborted Status. This bit is set to 1 if the user interrupts the current transmission by writing the LIN_CR1->PROC_RESET bit and is reset to 0 when a valid frame header is detected	0
3	DATA_REQ	RO	Data Request Status. When the frame header is received and the PID is correct, this bit is set to 1 to inform the controller that a new frame needs to be processed. The controller needs to determine the transmission direction, check type, transmission data, etc. of the current frame according to the PID. After the judgment is completed, the LIN_CR1->DATA_ACK bit is written to inform the controller to continue to receive or send the current frame	0
2	WKUP_TX_STATUS	RO	Wake Up Status. This bit is set after the controller starts to send the wake-up signal, and it will automatically clear 0 after the transmission is completed	0
1	TC	RO	End of Transmission Status. This bit is set after the frame is sent, and 0 is	0

			cleared after the frame header is received	
0	FC	RO	End of Frame Status. This bit is set after the frame is sent/received. Write 1 clear.	0

LIN_IER: Address(0x10)

Bit	Symbol	R/W	Description	Default
31:6	Reserved	RO	Not used	0
5	TXD_TIMEOUT_IER	RW	TXD Timeout Interrupt Enable. 0: Disable 1: Enable	0
4	WKUP_TIMEOUT_IER	RW	Wake-Up Timeout Interrupt Enable. 0: Disable 1: Enable	0
3	TXD_ERR_IER	RW	TXD Compare Error Interrupt Enable. 0: Disable 1: Enable	0
2	BUS_TIMEOUT_IER	RW	Bus Timeout Interrupt Enable. 0: Disable 1: Enable	0
1	DATA_REQ_IER	RW	Data Request Interrupt Enable. 0: Disable 1: Enable	0
0	FC_IER	RW	End of Frame Transmission Interrupt Enable. 0: Disable 1: Enable	0

LIN_CR1: Address(0x14)

Bit	Symbol	R/W	Description	Default
31:13	Reserved	RO	Not used	0
12	CHEK_SEL	RW	Checksum Select. 0: Classic 1: Enhanced	0
11:8	DATA_LENTH	RW	Data length. PID[5:4] length 2'b00 2 2'b01 2 2'b10 4 2'b11 8	0x0
7	PROC_RESET	RW	Frame Processing Reset. 0: Disable 1: Enable	0
6	SLEEP	RW	Sleep Request.	0

			0: Disable 1: Enable	
5	FRAME_DIR	RW	Frame Diretion Selection. 0: Disable 1: Enable	0
4	DATA_ACK	RW	Data Acknowledge. 0: Disable 1: Enable	0
3:2	Reserved	RO	Not Used	0
1	TRS_WKUP	RW	Transmit WakeUp Request. 0: Disable 1: Enable	0
0	Reserved	RO	Not used	0

LIN_CR2: Address(0x18)

Bit	Symbol	R/W	Description	Default
31:9	Reserved	RO	Not used	0
8	AUTO_RATE_EN	RW	Auto BaudRate Enable. 0: Disable 1: Enable	1
7:4	Reserved	RO	Not Used	0
3	TXD_ERR_MODE	RW	TXD Compare Err Mode. 0: Disable 1: Enable	0
2	TXD_TIMEOUT_DET_EN	RW	TXD timeout Detect Enable. 0: Disable 1: Enable	1
1	TXD_MONIT_EN	RW	TXD Monitor Compare Enable. 0: Disable 1: Enable	0
0	AUTO_SLEEP_EN	RW	Auto Sleep Enable. 0: Disable 1: Enable	1

LIN_RTR: Address(0x1C)

Bit	Symbol	R/W	Description	Default
31:23	Reserved	RO	Not used	0
22:8	DIVIDER	RW	BaudRate Divider: $fclk=fsysclk/(divider+fraction/256)$	0x3FFF
7:0	FRACTION	RW	Rate Fraction Configuration: $fclk=fsysclk/(divider+fraction/256)$	0x0

LIN_TSR: Address(0x20)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:26	Reserved	RO	Not used	0
25:24	WKUP_LEN	RW	Length of wake-up signal 00: 250μs 01: 500μs 10: 750μs 11: 1000μs	0x1
23:19	Reserved	RO	Not used	0
18:16	INTERBYTE_LEN	RW	Inter-Byte length.	0x0
15:10	Reserved	RO	Not used	0
9:8	BUS_TIMEOUT_DET_THRES	RW	LIN Bus timeout threshold. 00: 4s 01: 6s 10: 8s 11: 10s	0x0
7:2	WKUP_REC_THRES	RW	Wake Up Detected threshold. threshold = $WKUP\_THRES*T\_(lf\_clk)$	0x26
1:0	WKUP_REPEAT_THRES	RW	Wake Up Signal re-send threshold. If the controller does not receive a valid frame header within a certain time after sending the wake-up signal, it means that the wake-up has failed and the controller needs to send the wake-up signal again 00: 180ms 01: 200ms 10: 220ms 11: 240ms	0x0

LIN_DR3: Address(0x24)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:9	Reserved	RO	Not used	0
8	SWCHK_EN	RW	Software Checksum Calculate Enable. 0: Disable 1: Enable	0
7:0	DATA9	RW	data9.	0x0

LIN_PHY_ANR: Address(0x80)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:17	Reserved	RO	Not used	0
16	TXD_INVERT_EN	RW	LIN_TXD inversion enabled. 0: When LIN_TXD is 0, LIN sends 0, and when LIN_TXD is 1, LIN sends 1	1

			1: When LIN_TXD is 0, LIN sends 1, and when LIN_TXD is 1, LIN sends 0	
15:8	Reserved	RO	Not used	0
7	PU30K_EN	RW	The 30K pull-up resistor of the LIN_PHY is enabled	1
6	TX_EN	RW	LIN_PHY TXD enable	0
5	RX_EN	RW	LIN_PHY RXD enable	0
4	TX_RISE_CTRL	RW	LIN_TXD rise slope selection 0: Slow rise 1: Rapid rise	0
3:2	TX_SLOPE	RW	LIN_TXD swing rate	0x0
1:0	TX_BOOST	RW	LIN pull-down drive current	0x0

LIN_PHY_CR: Address(0x84)

Bit	Symbol	R/W	Description	Default
31:10	Reserved	RO	Not used	0
9	RXD_NEG_FILTER_EN	RW	RXD rising edge filtering enable This bit is used to filter out burrs during RXD0->1, and the width of the burrs to be filtered is controlled by LIN_TSR->RXD_POS_THRES	0
8	RXD_POS_FILTER_EN	RW	RXD falling edge filtering enable This bit is used to filter out burrs during RXD1->0, and the width of the burrs to be filtered is controlled by LIN_TSR->NEG_POS_THRES	0
7:6	Reserved	RO	Not used	0
5	TXD_AUTOOFF_EN	RW	TXD Auto Turnoff Enable. 0: TXD is controlled by LIN_PHY_ANR->TX_EN 1: TXD is shut down automatically	0
4	PU30K_AUTOOFF_EN	RW	PULLUP 30K Resistor Auto Turnoff When enabled, the 30K pull-up resistor of the LIN_PHY is automatically turned off after the LIN bus dominant level timeout	1
3	Reserved	RO	Not used	0
2	AUTO_ADDRESS_AUTOOFF	RW	Auto-Addressing Function Auto Disable. After this position, automatic addressing is automatically turned off in the event of a LIN bus explicit level timeout	1
1	AUTO_ADDRESS_LOCK	RW	Automatic addressing is locked After this position, the auto-addressing register cannot be changed	0
0	AUTO_ADDRESS_EN	RW	Automatic addressing enable	1

LIN_PHY_TSR: Address(0x88)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:16	Reserved	RO	Not used	0
15:8	RXD_NEG_THRES	RW	RXD Negedge Giltch Filter Threshold: threshold = $(RXD\_NEG\_THRES + 2)*T\_sysclk$	0x0
7:0	RXD_POS_THRES	RW	RXD Posedge Giltch Filter Threshold: threshold = $(RXD\_POS\_THRES + 2)*T\_sysclk$	0x0

LIN_M_BASE : 0x50000600

LIN_PHY_ANR: Address(0x80)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:17	Reserved	RO	Not used	0
16	TXD_INVERT_EN	RW	LIN_TXD inversion enabled. 0: When LIN_TXD is 0, LIN sends 0, and when LIN_TXD is 1, LIN sends 1 1: When LIN_TXD is 0, LIN sends 1, and when LIN_TXD is 1, LIN sends 0	1
15:9	Reserved	RO	Not used	0
8	PU1K_EN	RW	The 1K pull-up resistor of the LIN_PHY is enabled	1
7	PU30K_EN	RW	The 30K pull-up resistor of the LIN_PHY is enabled	0
6	TX_EN	RW	LIN_PHY TXD enable	0
5	RX_EN	RW	LIN_PHY RXD enable	0
4	TX_RISE_CTRL	RW	LIN_TXD rise slope selection 0: Slow rise 1: Rapid rise	0
3:2	TX_SLOPE	RW	LIN_TXD swing rate	0x0
1:0	TX_BOOST	RW	LIN pull-down drive current	0x0

LIN_PHY_CR: Address(0x84)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:6	Reserved	RO	Not used	0
5	TXD_AUTOOFF_EN	RW	TXD Auto Turnoff Enable. After this position, when the chip is in hibernate mode and the LIN bus is invisible, LIN_TXD will be automatically turned off 0: TXD is controlled by LIN_PHY_ANR->TX_EN 1: TXD is shut down automatically	0
4:0	Reserved	RO	Not used	0

WWDG

OVERVIEW

Window watchdog (WWDG) is usually used to monitor abnormal late or early operations of applications caused by external interference or unforeseeable logical conditions. Unless the countdown counter is refreshed before reaching 0x40, the watchdog circuit will generate an MCU reset when the preset time period is reached.

Before the countdown counter reaches the window register value, if the countdown counter valueis refreshed, an MCU reset will also be generated. This indicates that the countdown counter needs to be refreshed within a limited time window.

FEATURES

- Programmable free running decrement counter.
- Conditional reset:
  - When the value of the decrement counter is less than 0x40, if the watchdog is started, a reset is generated.
  - When the decrement counter is reloaded outside the window, if the watchdog is started, a reset is generated.
- If watchdog is enabled and interrupts are allowed, an Early wake-up interrupt (EWI) is generated when the decrement counter is equal to 0x40, which can be used to reload the counter to avoid a WWDG reset.

FUNCTIONAL DESCRIPTION

Timeout reset function

If the watchdog is enabled (write 1 into WDGA in WWDG_CR), and when the 7-bit decrement counter (T[6:0]) is flipped from 0x40 to 0x3F(T6 bit reset), a reset is produced.

A reset is also generated if the software reloads the counter when the value is greater than the number in the window register.

The timeout reset timing diagram of WWDG is shown as follows:

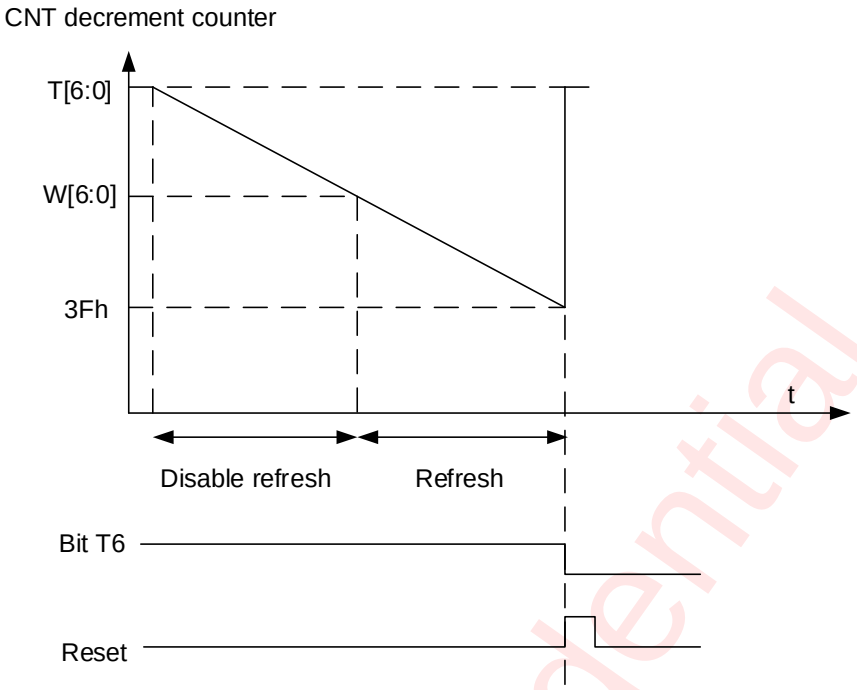


Figure 10 Time-out reset timing diagram of WWDG

The formula for calculating the timeout is as follows:

$$T_{WWDG} = T_{fclk} * 32 * 2^{WDGTB} * (T[5:0] + 1)$$

Min-max timeout values for fclk=16KHz

WDGTB	Min timeout	Max timeout
0	2ms	128ms
1	4ms	256ms
2	8ms	512ms
3	16ms	1.024s
4	32ms	2.048s
5	64ms	4.096s
6	128ms	8.192s
7	256ms	16.384s

Early wake-up interrupt function

Setting the EWI bit in the WWDG_CFR register can enable the early wake-up interrupt, which is generated when the decrementing counter reaches 0x40. The corresponding interrupt service routine (ISR) can be used to load the counter to prevent WWDG from resetting.

Clear the EWI interrupt by writing '0' to the EWIF bit in the WWDG_SR register.

REGISTER LIST

Register Mapping

WWDG_BASE : 0x50000300

Offset	Register Name	Description	Reset Value
0x00	WWDG_CR	WWDG Module control register	0x7F

0x04	WWDG_CFR	WWDG configuration register	0x1FF
0x08	WWDG_SR	WWDG status register	0x00

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write.

WWDG_CR: Address(0x00)				
Bit	Symbol	R/W	Description	Default
31:8	Reserved	RO	Not used	0
7	WDGA	RW	WWDG enable bit 0: Disable 1: Enable	0
6:0	T	RW	7-bit counter	0x7F

WWDG_CFR: Address(0x04)				
Bit	**Symbol**	**R/W**	**Description**	**Default**
31:11	Reserved	RO	Not used	0
10	EWI	RW	Early Wakeup Interrupt 1: When the counter reaches 0x40, an interrupt is generated. This interrupt can only be cleared by the hardware reset	0
9:7	WDGTB	RW	Time-base prescaler 000: FCLK/32/1, timing period: 2ms, maximum timeout is 2ms*(0x7F-0x3F)=128ms 001: FCLK/32/2, timing period: 4ms, maximum timeout is 4ms*(0x7F-0x3F)=256ms 010: FCLK/32/4, timing period: 8ms, maximum timeout is 8ms*(0x7F-0x3F)=512ms 011: FCLK/32/8, timing period: 16ms, maximum timeout is 16ms*(0x7F-0x3F)=1.024s 100: FCLK/32/16, timing period: 32ms, maximum timeout is 32ms*(0x7F-0x3F)=2.048s 101: FCLK/32/32, timing period: 64ms, maximum timeout is 64ms*(0x7F-0x3F)=4.096s 110: FCLK/32/64, timing period: 128ms, maximum timeout is 128ms*(0x7F-0x3F)=8.192s 111: FCLK/32/128, timing period: 256ms, maximum timeout is 256ms*(0x7F-0x3F)=16.384s	0x3
6:0	W	RW	7-bit window value These bits contain the window value to compare with the decrement counter	0x7F

WWDG_SR: Address(0x08)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	RO	Not used	0
0	EWIF	RW	early wakeup interrupt flag When the counter value reaches 40h, this bit is set '1' by the hardware. It must be cleared by software writing '0'.	0

WAKEUP_TIMER

FEATURES

- 25-bit counter, operating in 16kHz clock domain.
- Support to wake up the chip from low power mode.
- Support interrupt control function, including interrupt generation, interrupt clearing, and interrupt masking.
- When the timer is working, TimerLoadCount changes. It will only be updated by the timer when it reaches 0 or when it is restarted. Otherwise, the timer will continue to decrease until it reaches 0 before updating TimerLoadCount.

CONFIGURATION FLOW

1) Configure WAKEUP_TIMER loading value

The WAKEUP_TIMER reload value can be configured by the WAKEUP_TIMER.TimerLoaderCount.

2) Configure WAKEUP_TIMER interrupt

The activation and masking of WAKEUP_TIMER interrupts can be configured by the WAKEUP_TIMER. TimerControlReg.

3) Enable WAKEUP_TIMER interrupt

Enable WAKEUP_TIMER global interrupt and set interrupt priority based on application configuration.

4) Start WAKEUP_TIMER

The enable and disable of WAKEUP_TIMER can be configured by the WAKEUP_TIMER.TimerControlReg.

5) WAKEUP_TIMER interrupt generated

When WAKEUP_TIMER is enabled, the counter starts counting down from the value configured by the WAKEUP_TIMER.TimerLoaderCount register. When the count value decreases to 0, a WAKEUP_TIMER interrupt is generated, the counter value returns to WAKEUP_TIMER, and start counting down again based on the value configured in the TimerLoaderCount.

6) WAKEUP_TIMER interrupt clearing

After the WAKEUP_TIMER interrupt is generated, it is necessary to write to the WAKEUP_TIMER.TimerEOI register to clear the WAKEUP_TIMER interrupt status.

7) Reconfigure WAKEUP_TIMER count value

During the counting process of WAKEUP_TIMER, the register WAKEUP_TIMER.TimerLoadCount can be configured at any time to update the load count value. At this time, the original counting process is not interrupted. When the original counting process is decremented to 0, the updated load count value is returned to continue decrementing.

8) Disable WAKEUP_TIMER

During the use of WAKEUP_TIMER, the WAKEUP_TIMER.TimerControlReg register can be configured at any time to disable WAKEUP_TIMER.

USAGE RESTRICTIONS

1) TimerLoadCount must be set before starting timer_en, and if TimerLoadCount is modified midway, Timer must be disabled and then enable again.

2) TimerLoadCount time interval cannot be set to 0x1~0xF, which is a relatively small value. It is desired to set the time to be at least 1ms, TimerLoadCount>0x1F.

REGISTER LIST

Register Mapping

WAKEUP_TIMER_BASE : 0x50000E00

Offset	Register Name	Description	Reset Value
0x00	TimerLoadCount	Wakeup timer load counter register	0x1FFFFFF
0x08	TimerControlReg	Wakeup timer control register	0
0x0c	TimerEOI	Write 1 Clear Interrupt Status register	0
0x10	TimerIntStat	Interrupt status register	0

Register Detailed Description

RO: Read only, W: Write only, RW: Read/Write, WC: Write clear.

TimerLoadCount: Address(0x00)

Bit	Symbol	R/W	Description	Default
31:25	Reserved	RO	Not used	0
24:0	TimerLoadCount	RW	Wakeup timer load counter register	0x1FFFFFF

TimerControlReg: Address(0x08)

Bit	Symbol	R/W	Description	Default
31:3	Reserved	RO	Not used	0
2	TimerIntMsk	RW	interrupt mask. 0: Diasble 1: Enable	0
1	Reserved	RO	Not used	0
0	TimerEn	RW	Timer enable. 0: Diasble 1: Enable	0

TimerEOI: Address(0x0c)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	RO	Not used	0
0	TimerIntclr	WC	Write 1 Clear Interrupt Status.	0

TimerIntStat: Address(0x10)

Bit	Symbol	R/W	Description	Default
31:1	Reserved	RO	Not used	0
0	TimerIntstate	RO	Interrupt state.	0

APPLICATION INFORMATION

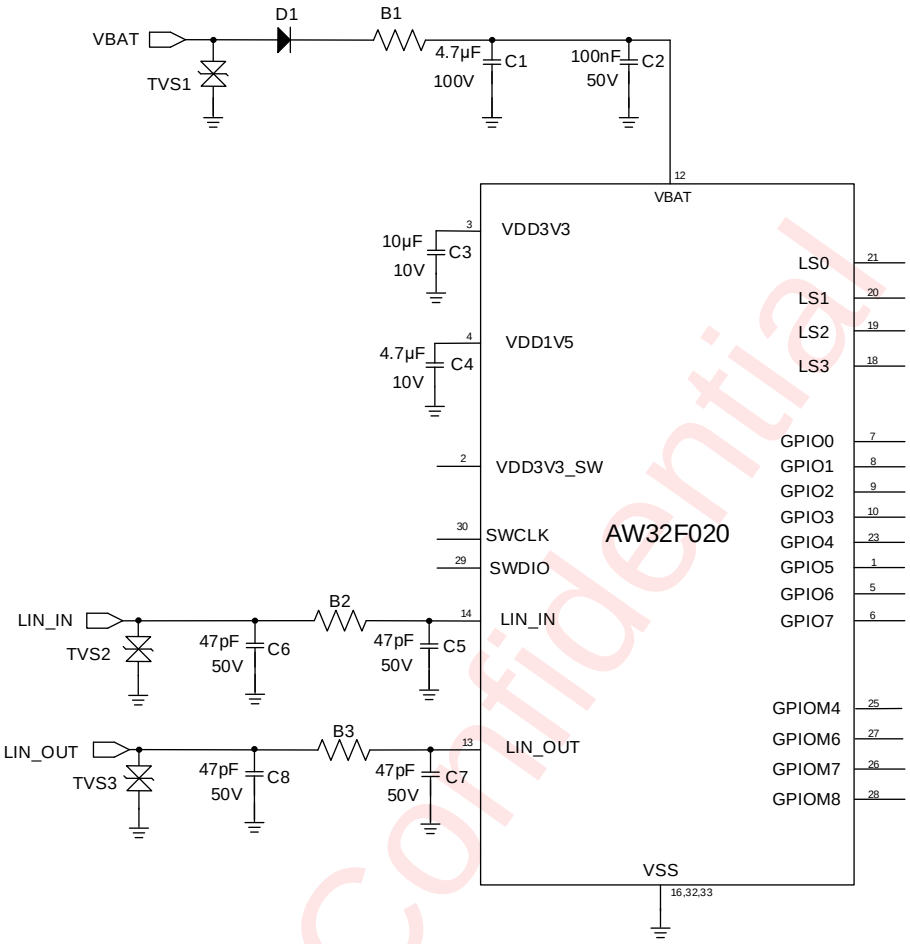


Figure 11 AW32F020QNR-Q1 Typical Application Circuit

COMPONENT SELECTION INSTRUCTIONS

1. The recommended value of capacitor C1 is 4.7μF, C2 is 0.1μF, C3 is 10μF, C4 is 4.7μF, and C5, C6, C7 and C8 is 47pF. The specific capacity can be selected according to the application and debugging.
2. VBAT, LIN_IN and LIN_OUT lines reserve series magnetic beads B1, B2 and B3, mainly to reduce EMI interference, which can be selected according to debugging. It is recommended that to place B1 between D1 and C1.
3. The series diode is recommended at the input of the VBAT power supply, and the circuit to prevent the reverse current is realized by using the unidirectional conductive characteristics of the diode.
4. VBAT, LIN_IN and LIN_OUT wiring port reserved TVS tube, recommended bipolar working voltage of 27V bipolar TVS tube, improve ESD capacity, specific specifications can be selected according to the application and debugging.
5. LS0~3 are 4-channel low-side switches with open-drain output, supporting 9-bit resolution PWM output and a maximum drive current of 60mA, which can be used for driving relays. LS0~3 can be configured as ADC detection pins. Unused LSx pins should be left floating.
6. AW32F020 has a total of 12 I/O pins, including GPIO0~GPIO7, GPIOM4, 6, 7, and 8, with direction control for input and output. Each IO port supports dual-edge interrupt triggering, clearing, and status querying. The maximum drive current of GPIO ports is 10mA;
7. The functions of GPIOs are disabled by default and should be configured according to actual needs; among them, GPIOMs can be reused for hardware I2C, SPI, UART, and TIM3 output; GPIO0~5 can be

configured as ADC detection pins; unused GPIO ports are recommended to be left floating

8. When programming the firmware by SWD, VDD3V3 can be used to power the chip, with a recommended power supply range of 3.2~5.5V

COMPONENT LIST

Component	Designator	Description	Range	Recommend Value	Unit
Capacitance	C1	100V, X7R	1~100	4.7	μF
Capacitance	C2	50V, X7R	/	0.1	μF
Capacitance	C3	10V, X7R	4.7~10	10	μF
Capacitance	C4	10V, X7R	1~4.7	4.7	μF
Capacitance	C5、C6、C7、C8	50V, X7R	/	220	pF
Diode	D1		/	/	/
TVS	TVS1、TVS2、TVS3	PESD1IVN27-A	/	/	/
Bead	B1, B2, B3	MPZ1608	/	/	/

PCB LAYOUT CONSIDERATION

To fully utilize the performance of the AW32F020QNR-Q1, the PCB layout and routing should be carefully considered. The design process should follow the following principles:

1) The power decoupling capacitors (C1 C2) of chip AW32F020QNR-Q1 should be placed as close as possible to the VBAT pin of the chip, and follow the principle that the current first passes through the large capacitor, then passes through the small capacitor, and finally reaches the chip. It is recommended to place ground around the chip, away from high-frequency devices.

2) The wire width of VBAT power supply to chip pin is not less than 0.25mm.

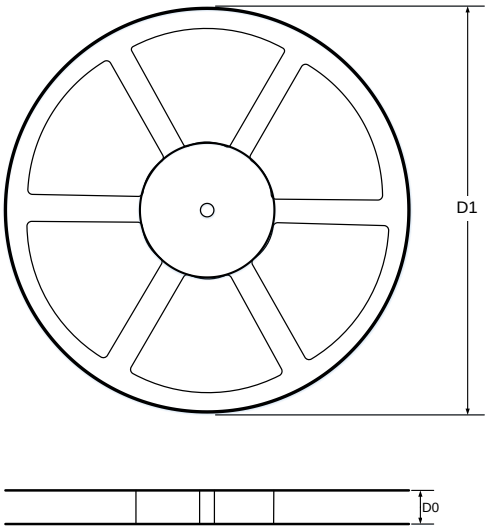
3) Requirements for LIN_IN and LIN_OUT:

- The signal lines of LIN_IN and LIN_OUT are separated from the power line to avoid crossing with the power line to reduce interference.
- Separate layout with high-frequency signal lines to reduce mutual interference.
- Chip away from the high-frequency chip at least 3mm.
- Reasonably set the direction and length of LIN_IN and LIN_OUT signal lines to ensure the stability and reliability of signal transmission.

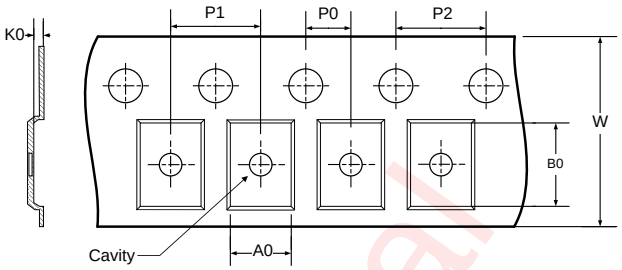
4) The maximum output current of GPIOs port is 10mA, and the recommended trace width is not less than 0.25mm.

Tape And Reel Information

REEL DIMENSIONS

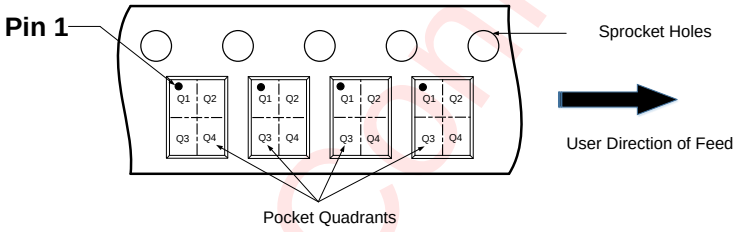


TAPE DIMENSIONS



A0：Dimension designed to accommodate the component width
B0：Dimension designed to accommodate the component length
K0：Dimension designed to accommodate the component thickness
W：Overall width of the carrier tape
P0：Pitch between successive cavity centers and sprocket hole
P1：Pitch between successive cavity centers
P2：Pitch between sprocket hole
D1：Reel Diameter
D0：Reel Width

QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



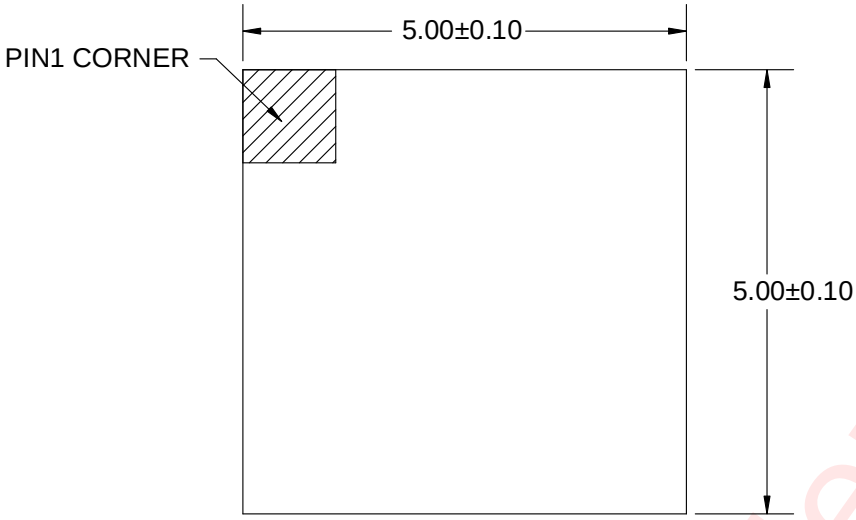
Note：The above picture is for reference only. Please refer to the value in the table below for the actual size

DIMENSIONS AND PIN1 ORIENTATION

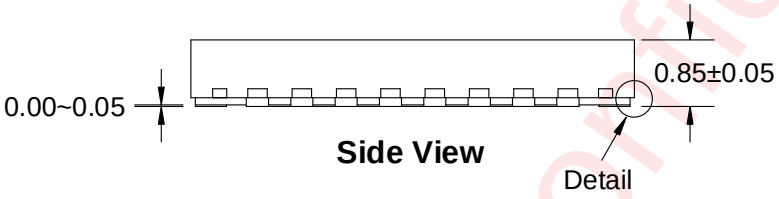
D1 (mm)	D0 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P0 (mm)	P1 (mm)	P2 (mm)	W (mm)	Pin1 Quadrant
330	12.4	5.3	5.3	1.1	2	8	4	12	Q1

All dimensions are nominal

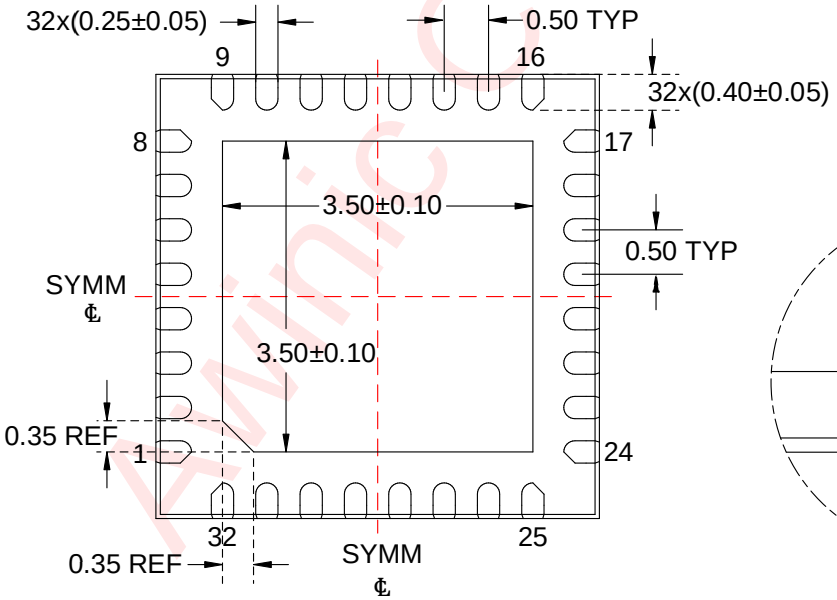
Package Description



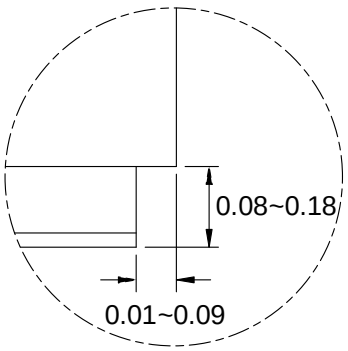
Top View



Side View



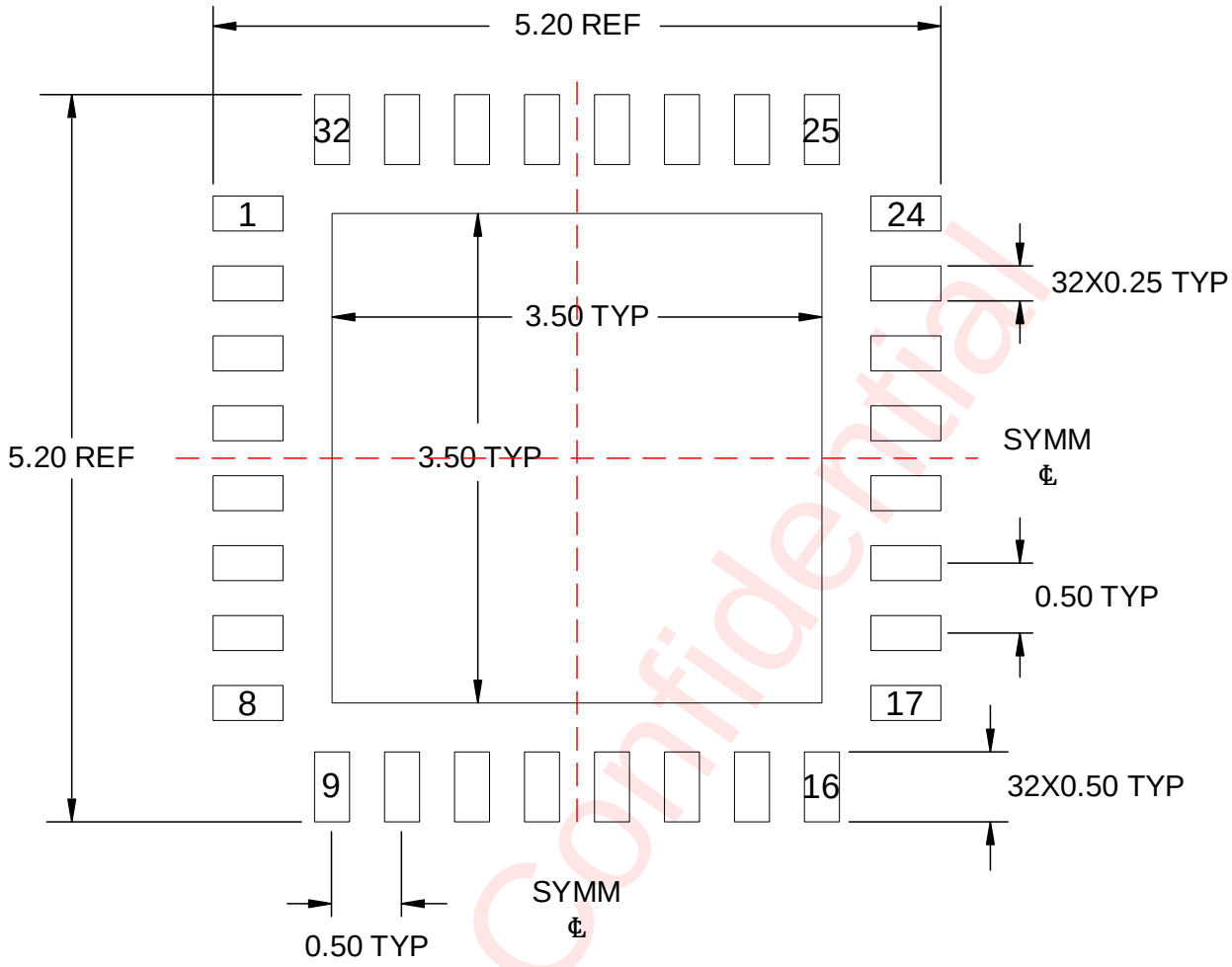
Bottom View

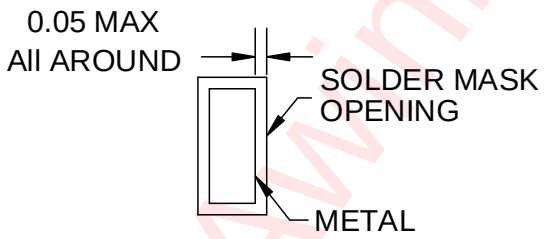


Detail

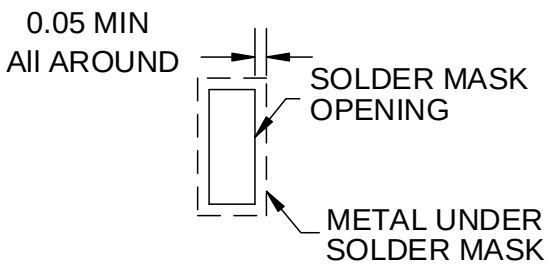
Unit:mm

Land Pattern Data





NON-SOLDER MASK DEFINED



SOLDER MASK DEFINED

Unit：mm

Revision History

Version	Date	Change Record
V1.0	Aug. 2025	Officially released

Disclaimer

All trademarks are the property of their respective owners. Information in this document is believed to be accurate and reliable. However, Shanghai AWINIC Technology Co., Ltd (AWINIC Technology) does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

AWINIC Technology reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. Customers shall obtain the latest relevant information before placing orders and shall verify that such information is current and complete. This document supersedes and replaces all information supplied prior to the publication hereof.

AWINIC Technology products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of an AWINIC Technology product can reasonably be expected to result in personal injury, death or severe property or environmental damage. AWINIC Technology accepts no liability for inclusion and/or use of AWINIC Technology products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications that are described herein for any of these products are for illustrative purposes only. AWINIC Technology makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

All products are sold subject to the general terms and conditions of commercial sale supplied at the time of order acknowledgement.

Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Reproduction of AWINIC information in AWINIC data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. AWINIC is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of AWINIC components or services with statements different from or beyond the parameters stated by AWINIC for that component or service voids all express and any implied warranties for the associated AWINIC component or service and is an unfair and deceptive business practice. AWINIC is not responsible or liable for any such statements.

