Functional overview

- High-performance low-power **8** -bit **LGT8XM** core
- Advanced **RISC** architecture
  - 131 instructions, more than 80% of the implementation of a single cycle
  - 32x8 general purpose working registers
  - 16MHz work up to 16MIPS implementation efficiency
  - Internal single cycle multiplier (8x8)

Nonvolatile program and data storage space

  - 4K / 8K / 16K / 32Kbytes on-chip programming **FLASH** program memory
  - 512 / 1K / 1K / 2Kbytes internal **SRAM**
  - Programmable E2PROM analog interface for byte access
  - A new program encryption algorithm to ensure user code security

- Peripheral controller
  - Two 8-bit timers with independent prescaler support compare output mode
  - A 16-bit timer with independent prescaler that supports input capture and compare output
  - The internal 32KHz calibrates the **RC** oscillator to implement the real-time counter function
  - Up to 6 **PWM** outputs can be supported , programmable dead zone control
  - 8-channel 12-bit high-speed analog-to-digital converter (**ADC**)
  - Two analog comparators (**ACs**) that support expansion from the ADC input channels
  - Two fixed-gain operational amplifiers (**OPA**) that can be used as front-end inputs for ADC / AC
  - Internal 1.25V / 2.56V ± 1% calibrable reference voltage source
  - Two 8-bit DACs that can be used to generate a reference voltage source
  - Programmable Watchdog Timer (**WDT**)
  - Programmable Synchronous / Asynchronous Serial Interface (**USART**)
  - Synchronous peripheral interface (**SPI**), programmable master / slave operating mode
  - Programmable two-wire serial interface (**TWI**), compatible with **I2C** master-slave mode
- Special processor function
  - **SWD** two-wire debugging / production interface
  - External interrupt source and I / O level change interrupt support
  - Built-in power-on reset circuit (**POR**) and 3 low-voltage detection circuit (**LVD**)
  - Built-in 1% calibrable 32MHz RC oscillator
  - Built-in 1% calibrable 32KHz RC oscillator
  - External support 32.768KHz and 400K ~ 20MHz crystal input
- **I / O** and package
  - QFP32L (up to 30 I / O)
  - SSOP28 / 24/20/16
- working environment
  - Working voltage: 1.8V ~ 5.5V
  - Operating frequency: 0 ~ 20MHz
  - Operating temperature: -40C ~ +85C
  - **HBM ESD** : **> ± 4000V**

8-bit LGT8XM

RISC Microcontroller with In-System Programmable FLASH Memory

LGT8F48D

LGT8F88D

LGT8F168D

LGT8F328D

Data book

Version 1.0.5

Application areas

Kitchen electric

- Induction Cooker
- Microwave oven
- Rice cookers and so on

home appliances

- Soymilk
- coffee pot
- Water heaters and so on

Intelligent control circuit

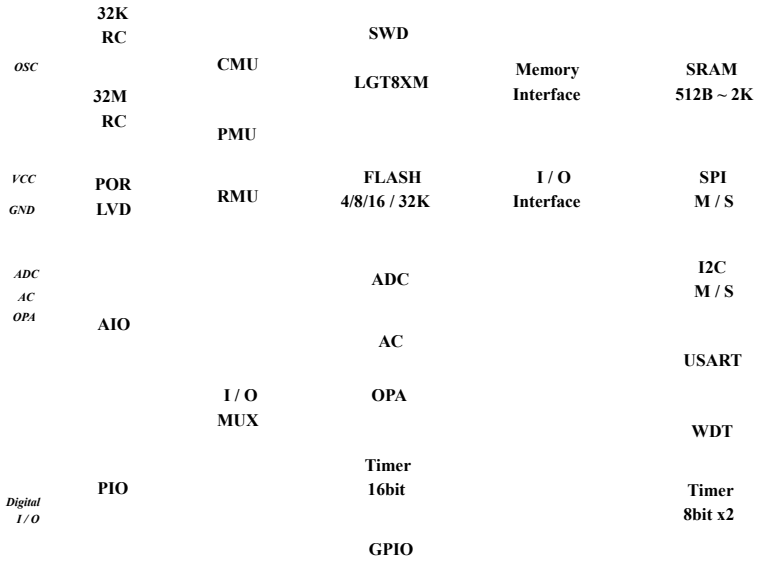
- Battery management
- Electric products
- Smart toys

Hand hold the instrument

Page 2

system framework

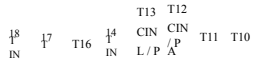
LGT8FX8D

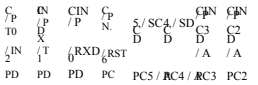


Module name	Module function
SWD	Debugging module, while achieving online debugging and ISP functions
LGT8XM	8bit high performance RISC kernel
CMU	The clock management module generates the various operating clocks required by the system
PMU	Power management module, responsible for managing the conversion between the working state of the system
RMU	Reset generation module
POR / LVD	Power-on reset module and low-voltage detection circuit
ADC	8-channel 12-bit analog-to-digital converter
AC	Analog comparator
OPA	Operational Amplifier
Timer	Timer / Event Counter
WDT	Watchdog reset module
SPI M / S	Master slave SPI controller
I2C M / S	Master slave I2C controller
USART	Synchronous / asynchronous serial transceiver
AIO	ADC input channel
PIO	Programmable digital I / O

Page 3

Package definition

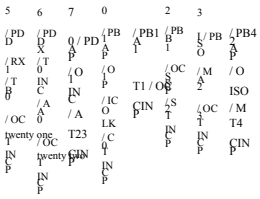




PCINT19 / OC2B / INT1 / PD3
PCINT20 / XCK / T0 / DAO / PD4
PCINT24 / OC0A / PE4
VCC
GND
PCINT25 / CLKO / DAO1 / PE5
PCINT6 / OSC1 / PB6
PCINT7 / OSC2 / PB7

LGT8FX8 Series
QFP32L

PC1 / ADC1 / PCINT9
PC0 / ADC0 / OC0A / PCINT8
PE3 / ADC7 / ACIN3
PE2 / SWD
PE6 / VREF
PE1 / ADC6 / ACIN2
PE0 / SWC
PB5 / OPA3 / SCK / PCINT6



PCINT14 / RSTN / PC6
PCINT16 / RXD / PD0
PCINT17 / TXD / PD1
PCINT18 / INT0 / PD2
PCINT19 / OC2B / INT1 / PD3
PCINT20 / XCK / T0 / DAO / PD4
VCC
GND
PCINT6 / OSC1 / PB6
PCINT7 / OSC2 / PB7
PCINT21 / OC0B / T1 / PD5
PCINT22 / OC0A / ACIN0 / PD6
PCINT23 / ACIN1 / OPA0 // PD7
PCINT0 / CLKO / ICP1 / OPA1 / PB0

LGT8FX8 Series
SSOP28

PC5 / ADC5 / SCL / PCINT13
PC4 / ADC4 / SDA / PCINT12
PC3 / ADC3 / PCINT11
PC2 / ADC2 / PCINT10
PC1 / ADC1 / PCINT9
PC0 / ADC0 / PCINT8
PE2 / ADC7 / ACIN3 / SWD
PE6 / AVREF
PE0 / ADC6 / ACIN2 / SWC
PB5 / OPA3 / SCK / PCINT5
PB4 / OPA2 / MISO / PCINT4
PB3 / MOSI / OC2A / PCINT3
PB2 / SPSS / OC1B / PCINT2
PB1 / OC1A / PCINT1

Page 4

RESETN / PC6
INT0 / PD2
INT1 / PD3
VCC
GND
OSC1 / PB6
OSC2 / PB7
OC0B / T1 / RXD / PD5

LGT8FX8 Series
SSOP24

PC5 / ADC5 / SCL
PC4 / ADC4 / SDA
PC3 / ADC3
PC2 / ADC2
PC1 / ADC1
PC0 / ADC0 / OC0A
PE2 / ADC7 / ACIN3 / VREF / SWD
PE0 / ADC6 / ACIN2 / SWC

OC0A / ACIN0 / TXD / PD6
ACIN1 / OPA0 / PD7
CLKO / ICP1 / OPA1 / PB0
OC1A / PB1
PB5 / OPA3 / SCK
PB4 / OPA2 / MISO
PB3 / MOSI / OC2A
PB2 / OC1B / SPSS

RESETN / PC6
INT0 / PD2
INT1 / PD3
VCC
GND
OSC1 / PB6
OSC2 / PB7
OC0B / T1 / RXD / PD5
OC0A / ACIN0 / TXD / PD6
ACIN1 / OPA0 / PD7

LGT8FX8 Series
SSOP20

PC5 / ADC5 / SCL
PC4 / ADC4 / SDA
PC1 / ADC1
PC0 / ADC0 / OC0A
PE2 / ADC7 / ACIN3 / VREF / SWD
PE0 / ADC6 / ACIN2 / SWC
PB5 / OPA3 / SCK
PB4 / OPA2 / MISO
PB3 / MOSI / OC1B / OC2A
PB1 / OPA1 / ICP1 / OC1A / SPSS

Package instructions

In the LGT8FX8D family of packages, the QFP32L package leads all pins. Other packages are bundled with multiple internal I / O on a QFP32 basis Pin generated on the pin. Special attention should be paid when configuring pin orientation. The following table lists the bindings for the various package pins:

QFP32L		DWPU	SSOP28L	SSOP24L	SSOP20L
01	**PD3 / INT1 / OC2B / PCINT19**	**-**	**05**	**03**	**03**
02	**PD4 / DAO / T0 / XCK / PCINT20**	**Y**	**06**	**-**	
03	**PE4 / 0C0A * / PCINT24**	**Y**	**-**	**-**	**-**
04	**VCC**	**-**	**07**	**04**	**04**
05	**VSS**	**-**	**08**	**05**	**05**
06	**PE5 / DAO1 / CLKO / PCINT25**	**Y**	**-**	**-**	**-**
07	**PB6 / OSC1 / PCINT6**	**Y (#)**	**09**	**06**	**06**
08	**PB7 / OSC2 / PCINT7**	**Y**	**10**	**07**	**07**
09	**PD5 / RXD * / T1 / 0C0B / PCINT21**	**-**	**11**	**08**	**08**
10	**PD6 / TXD * / ACIN0 / OC0A * / PCINT22**	**Y**	**12**	**09**	**09**
11	**PD7 / ACIN1 / PCINT23**	**Y**	**13**	**10**	**10**
12	**PB0 / ICP1 / CLKO * / PCINT0**	**Y**	**14**	**11**	**11**
13	**PB1 / OC1A / SPSS * / PCINT1**	**-**	**15**	**12**	
14	**PB2 / OC1B / SPSS / PCINT2**	**-**	**16**	**13**	**12**

15	**PB3 / MOSI / OC2A / PCINT13**	**Y**	**17**	**14**	
16	**PB4 / MISO / PCINT4**	**Y**	**18**	**15**	**13**
17	**PB5 / SCK / PCINT6**	**Y**	**19**	**16**	**14**
18	**PE0 / SWC**	**-**	**20**	**17**	**15**
19	**PE1 / ADC6 / ACIN2**	**Y**			
20	**PE6 / AVREF**	**Y**	**twenty one**		
twenty one	**PE2 / SWD**	**-**	**twenty two**	**18**	**16**
twenty two	**PE3 / ADC7 / ACIN3**	**Y**			
twenty three	**PC0 / ADC0 / OC0A * / PCINT8**	**Y**	**twenty three**	**19**	**17**
twenty four	**PC1 / ADC1 / PCINT9**	**Y**	**twenty four**	**20**	**18**
25	**PC2 / ADC2 / PCINT10**	**Y**	**25**	**twenty one**	**-**
26	**PC3 / ADC3 / PCINT11**	**Y**	**26**	**twenty two**	**-**
27	**PC4 / ADC4 / SDA / PCINT12**	**Y**	**27**	**twenty three**	**19**
28	**PC5 / ADC5 / SCL / PCINT13**	**Y**	**28**	**twenty four**	**20**
29	**PC6 / RSTN / PCINT14**	**Y**	**01**	**01**	**01**
30	**PD0 / RXD / PCINT16**	**Y**	**02**	**-**	**-**
31	**PD1 / TXD / PCINT17**	**Y**	**03**	**-**	**-**
32	**PD2 / INT0 / PCINT18**	**Y**	**04**	**02**	**02**

(*): * Marked with the pin function to change the function of the second optional position, you can set the relevant register;
OC0A is controlled by OC0C0 of PMXCR and OC0AS bit of TCCR0B; RXD / TXD is controlled by TDD6 / RDD6 of PMXCR
System; SPSS is controlled by PMB1 bit of PMXCR;

(#): If the PB6 pin is not used as an external crystal pin, use an external weak pull-down, the internal configuration bit is output low;

DWPU: pin default weak pull-up. When these pins are input I / O, there is a non-closed weak pull-up (80K or so)

Page 6

Pin description

Pin name	Functional description
VCC	System power supply (1.8V ~ 5.5V)
GND	Systematically
OSC1	External crystal input and output
OSC2	
RSTN	External Asynchronous Reset Input
RXD	Synchronous / asynchronous UART interface
TXD	
XCK	
INT0 / 1	External interrupt input, asynchronous wake source
OC0A / B	Timer 0 compare output (PWM0A / B)
OC1A / B	Timer 1 compare output (PWM1A / B)
OC2A / B	Timer 2 compare output (PWM2A / B)
SCL	TWI Two-Wire Data Interface (I2C)
SDA	
SCK	SPI interface
SPSS	
MISO	
MOSI	
T0	Timer 0 external clock input
T1	Timer 1 external clock input
ICP1	Timer 1 external capture input
SWD / SWC	SWD debug interface
PCINTX	Pin level change interrupt function
ADC7 ... 0	ADC input channel

DAO0 / 1	DAC output channel
VREF	ADC external reference voltage input
AIN0 / 1	Analog Comparator 0 External Input
AIN2 / 3	Analog Comparator 1 External Input
OPA0 / 1	Operational Amplifier 0 External Input
OPA2 / 3	Operational amplifier 1 external input
CLKO	System clock output
PB7 ... 0	Programmable I / O
PD7 ... 0	Programmable I / O
PC6 ... 0	Programmable I / O
PE6 ... 0	Programmable I / O

LGT8XM kernel

- Low power design

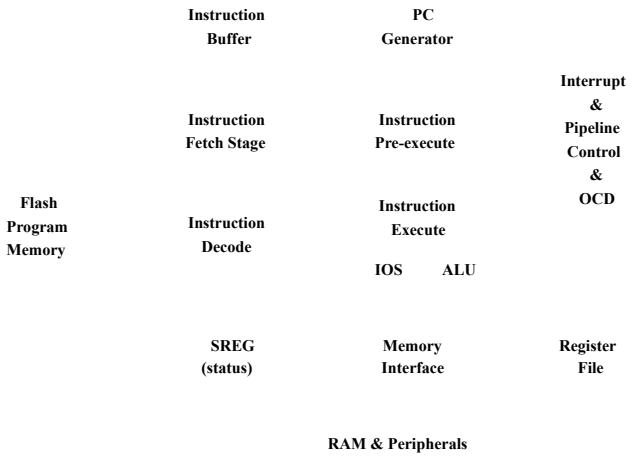
High efficiency RISC architecture

- 130 instructions, of which more than 80% for a single cycle
- Embedded online debugging (OCD) support

Overview

This section describes the LGT8XM kernel architecture and functionality. Kernel is the brain of the MCU, is responsible for ensuring that the program is positive The kernel must be able to perform calculations, control peripherals, and handle various interrupts.

The following figure shows the structure of the LGT8XM kernel:



To achieve greater efficiency and parallelism, the LGT8XM core uses a Harvard architecture - a separate data and program bus.
The instruction is executed through an optimized two-stage pipeline, and the two-stage pipeline can reduce the number of invalid instructions in the pipeline
The FLASH program memory access, so you can reduce the power consumption of the kernel. While the LGT8XM kernel in the fetch
So that the order of the increase in the instruction cache (which can cache two instructions), through the instruction cycle in the pre-execution module,
Further reducing the FLASH program memory access frequency; by our large number of tests, LGT8XM can be more than other similar
The architecture of the kernel to reduce the access to about 50% of the FLASH, greatly reducing the operating power of the entire system.

The LGT8XM core has 32 8-bit high-speed access to the common file register (Register file), help to achieve a single week

Period of arithmetic and logic operations (ALU). In general, the ALU operation of the two operands are from the general working register,
The result of the ALU operation is also written to the register file in one cycle.

32 of the 6 working registers are used to combine the two 16-bit registers, which can be used for indirect addressing
Address pointer, used to access external storage space and FLASH program space. LGT8XM supports single cycle 16-bit arithmetic
Count, greatly improve the efficiency of indirect addressing. The three special 16-bit registers in the LGT8XM core are named X, Y,
The Z register will be described later in detail.

ALU supports the arithmetic and logic operations between registers and between constants and registers. A single register operation can also be performed
Executed in ALU. After the ALU operation is complete, the effect of the operation result on the kernel state is updated to the status register (SREG).

Program flow control through the conditions and unconditional jump / call to achieve, can be addressed to the program area. most
The LGT8XM instruction is 16 bits. Each program address space corresponds to a 16-bit or 32-bit LGT8XM instruction.

After the kernel responds to an interrupt or subroutine call, the return address (PC) is stored on the stack. The stack is assigned to the system one
In the data SRAM, so the size of the stack is limited only by the size and usage of the SRAM in the system. All support interrupted or
Subroutine calls must be initialized by initializing the stack pointer register (SP), which can be accessed through IO space. data
SRAM can be accessed through 5 different addressing modes. LGT8XM's internal storage space is linearly mapped to one
Uniform address space. Please refer to the description of the storage section.

The LGT8XM core contains a flexible interrupt controller, which can be accessed via a status register
Board interrupt enable bit control. All interrupts have a separate interrupt vector. The interrupt priority is associated with the interrupt vector address
Corresponding relationship, the smaller the interrupt address, the higher the priority of the interrupt.

The I / O space contains 64 register spaces that can be addressed directly by the IN / OUT instruction. These registers are realistic
Core control, and status registers, SPI and other I / O peripherals control functions. This part of the space can be through IN / OUT
Direct access can also be accessed through the address they are mapped to the data memory space (0x20 - 0x5F). In addition,
The LGT8FX8D also includes extended I / O space, which is mapped to data storage space 0x60 - 0xFF, which can only be used
ST / STS / STD and LD / LDS / LDD instructions.

Arithmetic logic operation unit (ALU)

The LGT8XM contains a 16-bit arithmetic logic unit that can be completed in one cycle.
The arithmetic operation. The efficient ALU is connected to 32 general purpose working registers. Be able to complete two registers in one cycle
Or the arithmetic and logic operations between the register and the immediate data. ALU operations are divided into three kinds: arithmetic, logic and bit operations.
At the same time ALU part also contains a single cycle of the hardware multiplier, in a cycle to achieve two 8-bit register
Direct sign or unsigned operation. Please refer to the instruction set section for details.

Status register (SREG)

The status register mainly stores the result information generated by the last ALU operation. This information is used
Control the program execution flow. The status register is updated after the ALU operation is complete, thus eliminating the need to use the single
Unique comparison instructions can bring more compact and efficient code implementation.

The value of the status register is not automatically saved and restored when the response is interrupted and exited from the interrupt, which requires software to be
Now.

SREG register definition

SREG system status register

Address: 0x3F (0x5F) Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	I	T	H	S	V	N.	Z	C
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit definition

[0]	C	The carry flag indicates that the arithmetic or logic operation has caused the carry. For details, refer to the instruction description
[1]	Z	Zero flag, indicating that the result of arithmetic or logic operation is zero, refer to the instruction description section

[2]	N.	A negative sign indicates that a mathematical or logical operation produces a negative number, please refer to the instruction description section
[3]	V	The overflow flag indicates that the result of the two's complement operation has overflowed. Refer to the instruction description section
[4]	S	Symbol bit, equivalent to the exclusive OR operation of N and V, please refer to the instruction description section
[5]	H	The semi-carry flag, which is useful in the BCD operation, indicates that the byte operation produces a half carry
		Temporary bit, bit copy (BLD) and bit memory (BST) instructions, T bits will be used as a temporary
[6]	T	A value for temporarily storing a bit in a general purpose register. Please refer to the instruction description
		Minute
		The global interrupt enable bit must be set to 1 to enable the kernel to respond to an interrupt event. different
		The interrupt source is controlled by a separate control bit. The global interrupt enable bit is controlled by the interrupt signal entering the core
[7]	I	The last barrier. The I bit is automatically cleared by the hardware after the kernel responds to the interrupt vector, during execution
		After the return instruction (RETI) is set automatically. The I bit can also be changed using the SEI and CLI instructions
		Test instructions section

Generic working register

The general purpose register is optimized for the LGT8XM instruction set architecture. In order to achieve the efficiency and flexibility required for kernel execution, LGT8XM internal common working registers support several access modes:

An 8-bit read while an 8-bit write operation

- Two 8-bit read at the same time an 8-bit write operation
- Two 8-bit read at the same time a 16-bit write operation
- A 16-bit read while a 16-bit write operation

LGT8XM universal working register

	7	*0*	*Addr*	
	R0		***0x00***	
	R1		***0x01***	
	R2		***0x02***	
	...			
	R13		***0x0D***	
	R14		***0x0E***	
through	**R15**		***0x0F***	
use	**R16**		***0x10***	
work	**R17**		***0x11***	
For	**...**			
send	**R26**		***0x1A***	***X*** register low byte
Save	**R27**		***0x1B***	***X*** register high byte
Device	**R28**		***0x1C***	***Y*** register low byte
	R29		***0x1D***	***Y*** register high byte
	R30		***0x1E***	***Z*** register low byte
	R31		***0x1F***	***Z*** register high byte

Most instructions have direct access to all common working registers, and most of them are single-cycle instructions.

As shown in the figure above, each register corresponds to the address of a data memory space, which is mapped to

Data storage space. As soon as they do not really exist in the SRAM, but this unified mapping of the storage organization to visit
They brought great flexibility. The X / Y / Z register can be indexed into any general register as a pointer.

X / Y / Z register

The registers R26 ... R31 can be combined in two to form three 16-bit registers. These three 16-bit registers are mainly used for indirect
Addressing address pointer, X / Y / Z register structure is as follows:

	15	***XH***				***XL***	**0**
X register	**7**		**0**	**7**			**0**
	R27 (0x1B)			***R26 (0x1A)***			
	15	***YH***				***YL***	**0**
Y register	**7**		**0**	**7**			**0**
	R29 (0x1D)			***R28 (0x1C)***			
	15	***ZH***				***ZL***	**0**
Z register	**7**		**0**	**7**			**0**
	R31 (0x1F)			***R30 (0x1E)***			

In different addressing modes, these registers are used as fixed offset, auto increment and auto decremented address pointers,
For details, refer to the instruction description section.

Stack pointer

The stack is used to store temporary data, local variables, and return addresses for interrupts and subroutine calls. Need special attention
Yes, the stack is not designed to grow from a high address to a low address. The stack pointer register (SP) always points to the top of the stack. Stack
The pointer points to the physical space where the data SRAM is located, where the subroutine or interrupt call must hold the stack space. PUSH
The instruction will decrement the stack pointer.

The location of the stack in the SRAM must be set correctly by the software before the subroutine is executed or the interrupt is enabled. General situation
In this case, the stack pointer is initialized to the highest address of the SRAM. The stack pointer must be set to the high bit SRAM at the beginning
site. SRAM Refer to the system data storage section for the address of the system data storage map.

Stack pointer related to the instruction

instruction	Stack pointer	description
PUSH	Increase by 1	Data is pushed onto the stack
CALL	Increase by 2	The return address of the interrupt or subroutine call is pushed onto the stack
ICALL		
RCALL		
POP	Reduced by 1	The data is fetched from the stack
RET	Reduced by 2	The return address of the interrupt or subroutine call is removed from the stack
RETI		

The stack pointer consists of two 8-bit registers allocated in the I / O space. The actual length of the stack pointer matches the system implementation
turn off. In some chip implementations of the LGT8XM architecture, the data space is so small that only SPLs can satisfy addressing
In this case, the SPH register will not appear.

SPH / SPL Stack Pointer Register Definition

	SPH / SPL Stack Pointer Register
SPH: 0x3E (0x5E)	Default: RAMEND
SPL: 0x3D (0x5D)	

SP		SP [15: 0]
R / W		R / W
Initial		RAMEND

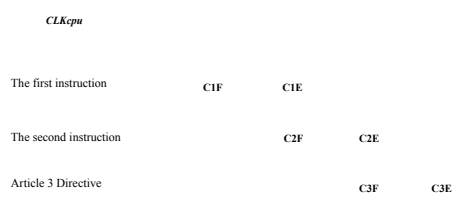
Bit definition

[7: 0]	SPL	The stack pointer is low for 8 bits
[15: 8]	SPH	The stack pointer is 8 bits high

Instruction execution timing

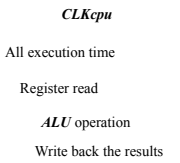
This section describes the general timing concepts for instruction execution. The LGT8XM kernel is driven by the kernel clock (CLKcpu)
The clock comes directly from the system with the clock source selection circuit.

The following figure shows the execution timing of the instruction pipeline based on the concept of the Harvard architecture and the fast access register file. This is to make
The kernel can get the physical guarantee of 1MIPS / MHz execution efficiency.



As can be seen from the above figure, the first instruction will be read during the implementation of the second instruction. When the second instruction goes into execution
Line period, while reading the third instruction at the same time. So that during the entire execution, there is no need to spend extra for reading instructions
Cycle, from the pipeline point of view, to achieve every Monday to implement the efficiency of a directive.

The following figure shows the access timing of the general working register. In one cycle, the ALU operation uses two registers as
Operand, and the ALU execution result is written to the destination register during this period.



Reset and interrupt handling

LGT8XM supports multiple interrupt sources. These interrupts and reset vectors in the program space correspond to a separate program
Volume entrance. In general, all interrupts have separate control bits. When the control bit is set, and enabled
After the kernel's global interrupt enable bit, the kernel can respond to this interrupt.

The lowest program space is retained by default as the reset and interrupt vector area. LGT8FX8D supports the complete interrupt list
Please refer to the description of the interrupt section. This list also determines the priority of the different interrupts. The lower the vector address is the interrupt,
The corresponding interrupt priority is higher. The reset (RESET) has the highest priority and then the INT0 - external interrupt request 0.
The start address of the interrupt vector table (except the reset vector) can be redefined to the beginning of any 256-byte alignment.
To be implemented by the IVSEL bit in the MCU control register (MCUCR) and the IVBASE vector base address register.

When the kernel response is interrupted, the global interrupt enable flag, I, is automatically cleared by hardware. The user can make the I bit by
Can achieve interrupt nesting. So that any subsequent interruption will interrupt the current interrupt service routine. I bit in the execution interrupt
After the return instruction (RETI) is set automatically, it can normally respond to subsequent interrupts.

There is a basic type of interrupt. The first type is triggered by an event, and the interrupt flag is set after an interrupt event occurs. for
This interrupt, the kernel response to the interrupt request, the current PC value is directly replaced by the actual interrupt vector address,

Line corresponding to the interrupt service subroutine, while the hardware automatically clear the interrupt flag. The interrupt flag can also be passed to the interrupt The position of the flag bit is cleared by 1. If the interrupt enable bit is cleared when an interrupt occurs, the interrupt flag bit will still be set To record an interrupt event. Wait until the interrupt is enabled, this record of the interrupt event will be immediately respond. Again, if in the interruption When executed, the global interrupt enable bit (SERG.I) is cleared and the corresponding interrupt flag bit is set to record the interrupt event, etc. When the global interrupt enable bit is set, these recorded interrupts will be executed in order of priority.

The second interrupt type is when the interrupt condition is always present, the interrupt is always responding. This interrupt does not require an interrupt flag Bit. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be acknowledged.

When the LGT8XM kernel exits from the interrupt service routine, the execution flow returns to the main program. In the main program After executing one or more instructions, you can respond to other waiting interrupt requests.

It should be noted that the system status register (SREG) does not automatically save after entering the interrupt service, The interrupt service is automatically restored after returning. It must be handled by the software.

Interrupts are disabled immediately when interrupts are disabled using the CLI instruction. After the CLI instruction occurs so the interrupt is both Will not get a response. Even if the interrupt is executed concurrently with the CLI instruction, it will not be responded. The following example says How to use the CLI to avoid interrupting the write sequence of the EEPROM:

Assembly code instance

```
IN R16, SREG ; store SREG value
CLI ; disable interrupts during timed sequence
SBI EECR, EEMPE ; start EEPROM write
SBI EECR, EEPE
OUT SREG, R16 ; restore SREG value (including I bit)
```


C language code instance

```
char cSREG
cSREG = SREG; / * store SREG value * /
/ * disable interrupts during timed sequence * /
_CLI ();
EECR | = (1 << EEMPE); / * start EEPROM write * /
EECR | = (1 << EEPE);
SREG = cSREG; / * restore SREG value (including I-bit) * /
```

When an interrupt is enabled with the SEI instruction, an instruction following the SEI instruction will first be preceded by an interrupt Is executed as an example of the following code:

Assembly code instance

```
SEI ; set Global Interrupt Enable
SLEEP ; enter sleep, waiting for interrupt
; note: will enter sleep before any pending interrupt (s)
```

C language code instance

```
__enable_interrupt (); / * set Global Interrupt Enable * /
__sleep (); / * enter sleep, waiting for interrupt * /
/ * note: will enter sleep before any pending interrupt (s) * /
```

Interrupt response time

The LGT8XM core is optimized for interruptions, making any interrupts available in the four system clock cycles response. After four system clock cycles, the interrupt service routine enters the execution cycle. In the four clocks, before the interruption

PC value is pushed onto the stack, the system execution process flow jumps to the interrupt vector corresponding to the interrupt service routine. If an interrupt occurs During a multi-cycle instruction execution, the kernel will ensure that the correct execution of the current instruction ends. If the interrupt occurs at the system In the sleep state (SLEEP), the interrupt response requires an additional 4 clock cycles. This increased clock cycle is used from the selection The synchronization period of the wake-up operation in sleep mode. For details of the sleep mode, refer to the relevant section of Power Management.

It takes 2 clock cycles to return from the interrupt service routine. In the two clock cycles, the PC is restored from the stack, Stack pointer plus 2, SREG (I) bit is set to 1.

Storage subsystem

Overview

This section describes the different memory cells within the LGT8FX8D family. The LGT8XM architecture supports two main ones Department of storage space, respectively, data storage space and program storage space. In addition, LGT8FX8D internal also contains the data FLASH, Through the internal controller can realize the EEPROM interface data storage function. In addition, LGT8FX8D system also contains A special storage unit for storing system configuration information and the global device number (GUID) of the chip.

LGT8FX8D series chip contains LGT8F48D / 88D / 168D / 328D four different models; four models outside Set and package is fully compatible, the difference is FLASH program storage space and internal data SRAM, the following table comparison Clearly described the LGT8FX8D series chip different storage space configuration:

model	FLASH	SRAM	E2PROM	Interrupt vector
LGT8F48D	4KB	512B	1KB	1 instruction word
LGT8F88D	8KB	1KB	2KB	1 instruction word
LGT8F168D	16KB	1KB	4KB	2 instruction words
LGT8F328D	32KB	2KB	Can be configured as 0K / 1K / 2K / 4K / 8K (Shared with FLASH)	2 instruction words

LGT8F328D is not used internally to simulate the E2 space of the E2PROM interface; for analog E2PROM Storage space and program FLASH share, the user can according to application requirements, select the appropriate configuration.

Due to the unique implementation of the analog E2PROM interface, the system requires twice the program FLASH space to simulate the E2PROM Storage space, such as for the LGT8F328D, if the user configured 1KB of E2PROM space, there will be 2KB bytes The program space is retained, leaving the 30KB of FLASH space for storing the program.

LGT8F328D program FLASH and E2PROM shared configuration table:

model	FLASH	E2PROM
	32KB	0KB
	30KB	1KB
LGT8F328D	28KB	2KB
	24KB	4KB

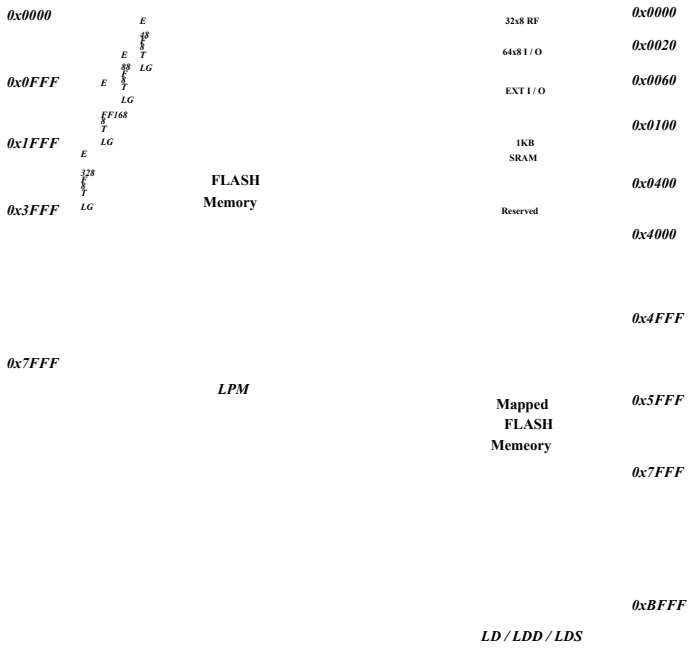
16KB 8KB

System programmable FLASH program storage unit

The LGT8FX8D family of microcontrollers internally includes 4K / 8K / 16K / 32K bytes of on-chip programmable programmable Sequential storage unit.

The program FLASH guarantees at least 20,000 erase cycles. LGT8FX8D internal integrated FLASH interface control Device, can be achieved in the system programming (ISP) and the program since the upgrade function. Please refer to this chapter for details on FLASH Description of the interface controller section.

Program space can also be accessed directly through the LPM instruction (read), this feature can be applied to the application of constant search table. At the same time FLASH program space is also mapped to the system data storage space, the user can also use LD / LDD / LDS real FLASH space is now on the visit. The program space is mapped to the address range starting from the data memory space 0x4000. As shown below:



SRAM data storage unit

The LGT8FX8D family of microcontrollers is a relatively complex microcontroller that supports a variety of different types of peripherals. Some peripheral controllers are allocated in 64 I / O register spaces. Can be accessed directly through the IN / OUT instruction. others Peripheral control register is allocated in the 0x60 ~ 0xFF area, because this part of the space is mapped to the data storage space, Can only be accessed via ST / STS / STD and LD / LDS / LDD instructions.

LGT8FX8D system data storage space from the 0 address, respectively, mapped a common working register file, I / O empty
Inter-expansion I / O space and internal data SRAM space. The beginning of the 32-byte address corresponds to the LGT8XM kernel 32

Generic working register. The next 64 addresses are standard I / O spaces that can be accessed directly through the IN / OUT instruction.
Then the 160 addresses are extended I / O space, followed by 1024 bytes of data SRAM. Starting at 0x4000
To 0xBFFF end of this part of the space, mapping FLASH program storage unit.

32 Universal working register	***0x0000 - 0x001F***
64 standard **I / O** space	***0x0020 - 0x005F***
160 Expand **I / O** space	***0x0060 - 0x00FF***
1024 **SRAM** data space	***0x0100 - 0x03FF***
Reserved	***0x0400 - 0x3FFF***
4K ~ 32KB **FLASH Memory**	***0x4000 - 0xBFFF***

The system data stores the unified mapping space

The system supports five different addressing modes that can cover the entire data space: direct access, indirect access with offset, between
Access, indirect access to the address before the visit, indirect access to the address after access. General working register R26 to
R31 is used for indirect access to address pointers. Indirect access can address the entire data storage space. Indirect with offset address
Asked to be addressed to 63 address spaces near the Y / Z register base address.

When using the register indirect access mode that supports auto-increment / decrement of address, the address register X / Y / Z will be used before the access occurs /
Automatically decremented / incremented by hardware. Please refer to the instruction set description section.

General I / O register

LGT8FX8D I / O space has three general-purpose I / O registers **GPIOR2 / 1/0** , these three registers can use IN / OUT means
Make access to user-defined data.

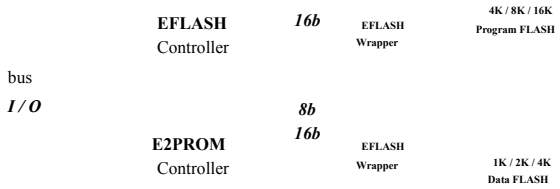
EFLASH / E2PROM interface controller

LGT8FX8D internal implementation of a flexible and reliable E2PROM interface controller, you can use the system has the number
According to FLASH storage space, to achieve byte access to read and write storage space, to achieve similar E2PROM storage applications; E2PROM
Interface simulation using erasure equalization algorithm, you can increase the data FLASH cycle of about 1 times, to ensure that 50,000
Times the erase cycle.

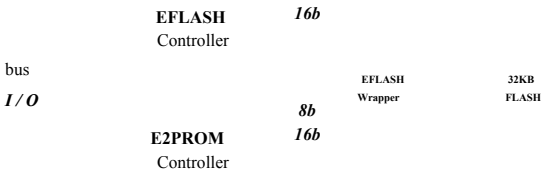
FLASH interface controller to achieve the FLASH program space on-line erase operation, you can achieve online through the software automatically
Upgrade the functionality of the firmware. Through the FLASH controller to access the program FLASH space, only supports 16-bit wide read and write access.

E2PROM and program FLASH space access details, please refer to the following detailed description.

LGT8F48D / 88D / 168D *FLASH / E2PROM* controller structure diagram



LGT8F328D *FLASH / E2PROM* controller structure diagram



FLASH / E2PROM interface read and write access

FLASH controller All controller registers can be accessed via I / O space. On the FLASH and E2PROM empty
Between the operation, but also through the configuration and control of these registers to achieve. Detailed use of the method will be described in the Register section
Separate instructions.

FLASH / E2PROM write access time, please refer to the table given later. FLASH controller can be automatically updated when
The state of the pre-operation, the user software can determine whether the current operation is completed by detecting these states, thereby starting the next word
Section of the operation. If the user code contains the FLASH / E2PROM operation, then you need to follow some principles. first
First, during power-up or power-down, V cc rises and falls slowly, causing the device to run for a long time at low voltage,
This will affect the current system to run the maximum frequency of the minimum voltage requirements, will also have an impact on the FLASH programming operation.
At this point you need to take the necessary protective measures. This will be described in detail in the next section.

In order to avoid misuse of the E2PROM, the operation of the E2PROM must follow a special process. Please refer to this
Section The final description of the EFLASH / E2PROM control register.

When the EFLASH / E2PROM is operated, the execution of the LGT8XM core will be held until the operation is complete.
To resume operation.

FLASH / E2PROM operation of the protection measures

If the VCC voltage is low, the data operation of the FLASH / E2PROM may cause an error due to the voltage being too low. with
Using the board-level E2PROM chip, you can use the same design.

FLASH / E2PROM data under low pressure operation errors may be caused by two reasons. First, a normal one

FLASH / E2PROM operation requires a minimum operating voltage, below this voltage, the operation will fail and cause data to occur error. The second reason is that the kernel is running at a certain frequency, it also requires a minimum voltage requirement, when below this Voltage, while the CPU and keep running at this frequency, will lead to the implementation of the instruction error, making FLASH / E2PROM The operation has failed.

You can avoid similar problems by the following simple method:

When the supply voltage is low, let the system into the reset state. This can be done by configuring the internal low voltage detection circuit (VDT) achieve. If VDT detects that the current operating voltage is below the set threshold, VDT will output a reset signal. in case VDT threshold can not meet the needs of the application, you can consider adding a reset circuit in the external.

I / O register space

For a detailed definition of I / O space, refer to the "Register Overview" section of the LGT8FX8D Data Sheet.

LGT8FX8D so the peripherals are assigned to the I / O space. All I / O space addresses can be LD / LDS / LDDD As well as ST / STS / STD instruction access. The accessed data is passed through 32 general purpose working registers. In 0x00 ~ 0x1F The I / O registers can be accessed by bit addressing instructions SBI and CBI. In these registers, the value of a bit can be To use the SBIS and SBIC instructions to detect the execution of the program. Please refer to the instruction set description section.

When using the IN / OUT instruction to access the I / O register, the address between 0x00 and 0x3F must be addressed. When using LD Or the ST instruction accesses the I / O space, the mapping address must be mapped through the I / O space in the system data memory Access (plus offset 0x20). Some other allocations in the extended I / O space of the peripheral registers (0x60 ~ 0xFF), only Access using the ST / STS / STD and LD / LDS / LDD instructions.

In order to be compatible with future devices, the reserved bit must be written to 0 when writing. Can not write on reserved I / O space operating.

Some registers include a status flag that needs to be written to 1 to clear. It should be noted that the CBI and SBI instructions Only support a specific bit, so CBI / SBI can only work in the register containing these status flags. In addition, CBI / SBI instructions can only work in the 0x00 to 0x1F address range of the register.

Register description

FLASH / E2PROM address register - EEARH / EEARL

	EEARH / EEARL
EEARH: 0x22 (0x42)	Default: 0x0000
EEARL: 0x21 (0x41)	
EEAR	EEAR [15: 0]
R / W	R / W
Initial value	0x0000

Bit definition

[7: 0]	EEARL EFLASH / E2PROM access address is low 8 bits.

[14: 8]	EEARH	EFLASH / E2PROM access address is high 7 digits
[15]	-	Keep not used

When using the EFLASH / E2PROM controller to access the program FLASH area, EEAR [14: 1] is used for access to 2 bytes Align the entire program space. EEAR [0] is only used when accessing the data register EEDR. Please refer to the following Description of the EEDR data register.
When using the EFLASH / E2PROM controller to access the data FLASH area (E2PROM), EEAR [12: 0] is used to access Maximum 8K bytes of E2PROM space. Access at this time supports 8/16 bit mode, no matter what mode, EEAR Are byte-aligned addressing.

EEAR allocation table:

model	FLASH	E2PROM	EEAR effective bit wide
LGT8F48D	4KB	1KB	EEAR [9: 0]
LGT8F88D	8KB	2KB	EEAR [10: 0]
LGT8F168D	16KB	4KB	EEAR [11: 0]
LGT8F328D	32/30/28/24 / 16K	0/1/2/4 / 8K	EEAR [12: 0]

FLASH / E2PROM Data Register - EEDR

EEDR - FLASH / E2PROM data register

EEDR: 0x20 (0x40)		Default: 0x00
EEDR		EEDR [7: 0]
R / W		R / W
Initial value		0x00
Bit definition		
[7: 0]	EEDR	EFLASH / E2PROM data register

important:

LGT8FX8D internal FLASH for the 16-bit interface, read / write data for the smallest unit of 16 bits. So FLASH control

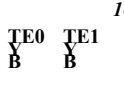
The internal data register is 16 bits. EEAR [0] is used to address the high and low 8 bits.

EEDR is an 8-bit wide data register whose meaning varies depending on the access pattern. When using FLASH When the controller accesses the internal program FLASH, the FLASH controller operates in 16-bit mode, and the EEDR The internal 16-bit data register interface will work with EEAR [0]. When using the FLASH controller to access data FLASH, FLASH controller access interface can work in 8/16-bit mode.

When working in 8-bit mode, EEDR is the actual data that needs to be read / written. EEAR [12: 0] is used to find Address up to 8K bytes of E2PROM space. The hardware will automatically complete the 8-bit data to 16-bit data access interface Conversion, the user does not need any additional operation.

When working with 16-bit mode, EEDR will also be used as an interface to access internal data that will work with EEAR [0] Work. In these two modes, the user needs to set the number of FLASH to be written by EEAR [0] and EEDR According to, or by EEAR [0] and EEDR, read out the required byte data.

The following figure illustrates the relationship between the I / O register EEAR / EEDR and the FLASH controller's internal interface:



EEAR [0]

EEDR

When using 8-bit mode, EEAR [12: 0] together with EEDR update the data of the specified byte position, the rest of the location
The data is automatically patched by the control logic inside the FLASH controller. Users do not have to care about specific implementations.
When using 16-bit mode, the user needs to update 16-bit data, that is, 2 bytes of data. Hardware according to EEAR [0]
To decide to update the upper 8 or 8 bits. Here's how to update your data:

```
OUT     EEARL, $ 0
OUT     EEDR, BYTE0
OUT     EEARL, $ 1
OUT     EEDR, BYTE1
# Set the programmed destination address
OUT     EEARL, ADDRL
OUT     EEARH, ADDRH
...
```


FLASH / E2PROM Configuration Register - ECCR (LGT8F328D specific)

ECCR - FLASH / E2PROM Configuration register (for LGT8F328D)

ECCR: 0x36 (0x56) Default: 0x00

ECCR	WEN	EEN	-	-	-	-	EC1	EC0
R / W	R / W	R / W	-	-	-	-	R / W	R / W
Initial value	0	0	0	0	0	0	0	0

Bit definition

[7]	WEN	ECCR write enable control Before modifying the ECCR, you must first write WEN 1 and then update it in 6 system cycles ECCR register contents
[6]	EEN	E2PROM enabled, only valid for LGT8F328D 1: Enable E2PROM analog interface, will be part of the space from 32KFLASH 0: disable E2PROM analog interface, 32KFLASH all for program space
[5: 2]	-	Keep not used
[1: 0]	EC [1: 0]	E2PROM space configuration 00: 1KB E2PROM, 30KB program FLASH 01: 2KB E2PROM, 28KB program FLASH 10: 4KB E2PROM, 24KB program FLASH 11: 8KB E2PROM, 16KB program FLASH

FLASH / E2PROM Controller Control Register - EECR

EECR - FLASH / E2PROM control register

EECR: 0x1F (0x3F) Default: 0x00

EECR	EEPM3	EEPM2	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W

Initial value	0	0	0	0	0	0	0	0

Bit definition

[7: 4]	EEPM [3: 0]	EFLASH / EPROM access mode control bit				
		[3]	[2]	[1]	[0]	Mode description
		0	0	0	x	8-bit mode read / write E2PROM (default)
		0	0	1	x	16-bit mode read / write E2PROM
		1	x	0	0	E2PROM erase (optional operation)
		1	x	0	1	Program FLASH erase (page erase)
		1	x	1	0	Program FLASH programming
		1	x	1	1	Reset the FLASH / E2PROM controller
[3]	EERIE	FLASH / E2PROM ready interrupt enable control. Write 1 enabled, write 0 disabled. When EEPE is After the hardware is automatically cleared, the E2PROM ready interrupt is active. During EPROM operation, Does not produce this interrupt				

[2]	EEMPE	FLASH / E2PROM programming operation enable control bits EEMPE is used to control whether EEPE is valid, when both EEMPE is set to 1 and EEPE is 0 After setting the EEPE to 1 for the next four cycles, the programming operation will be initiated. Otherwise programmed Invalid operation. After four cycles, EEMPE is automatically cleared
[1]	EEPE	FLASH / E2PROM programming operation enable bit
[0]	EERE	E2PROM read enable bit, the data will be valid after two system cycles

FLASH / E2PROM read and write control timing

The EECR register controls the implementation of all FLASH operations. Where EEPM mainly controls the operating mode and selection Type of operation. EEPM [3] The main choice is to operate the data FLASH (E2PROM) or program FLASH. When the operation is right Like the program FLASH, the data interface is fixed to 32-bit mode. When the operation target is data FLASH (E2PROM) You can choose a different data width. The default is 8-bit mode, this mode operation is the most simple and intuitive.

FLASH controller in the realization of E2PROM interface, the internal has been achieved when necessary to automatically erase the data FLASH Logic, so the EPROM erase command is optional, and this command is only used if the user needs to perform a separate erase.

EEMPE controls FLASH's erase / write timing, including program FLASH and E2PROM. Must be controlled at EEMPE Under the action to complete the response. EEPE can initiate all erase and program operations during EEMPE active timing. The specific type of operation is determined by EEPM [3: 0].

Read the E2PROM relatively simple, set the target address and mode, write EERE bit is the target address The corresponding 32-bit data is read into the FLASH controller, and the user can read the word of interest through the EEDR register Section. FLASH controller does not achieve the program FLASH space read operation, the user can easily use LPM or The program uses the LD / LDD / LDS instruction at the address of the data unified mapping space.

Data *FLASH / E2PROM* programming process examples

1. 8 -bit mode, programming ***E2PROM***

- Check the EEPE bit and wait for the FLASH controller to be idle
- Set the destination address to EEAR [8: 0]
- Set new data to EEDR
- Set EEPM [3: 1] = 000, EEPM [0] can be set to 0 or 1
- Set EEMPE = 1 while EEPE = 0
- Set the EEPE = 1 in four cycles

When the setting is complete, the FLASH controller will start the programming operation, during programming CPU will remain in the current instruction Address, until the operation is completed will continue to run. In the programming process, if you need to erase the data FLASH, The FLASH controller will automatically start the erase process.

2. 16 -bit mode, programming ***E2PROM***

- Check the EEPE bit and wait for the FLASH controller to be idle
- Set the 16-bit data by EEAR [0] and EEDR, refer to the EEDR register definition section

- Set the destination address to EEAR [12: 0]. Note that here is the byte alignment address, FLASH controller used EEAR [14: 1] as the address to access the FLASH.
- Set EEPM [3: 1] = 001, EEPM [0] can be set to 0 or 1
- Set EEMPE = 1 while EEPE = 0
- Set the EEPE = 1 in four cycles

Page 23

3. 8 -bit mode, read ***E2PROM***

- Check the EEPE bit and wait for the FLASH controller to be idle
- Set the destination address to EEAR [8: 0]
- Set EEPM [3: 1] = 000
- Set EERE = 1 to start E2PROM read operation
- Wait 2 cycles (perform two NOP operations)
- The data corresponding to the destination address is updated to the EEDR register

4. 16 -bit mode, read ***E2PROM***

- Detect the EEPE bit, waiting for the FLASH controller to be idle
- Set EEAR [12: 0] as the destination address and address 2 bytes
- Set EEPM [3: 1] = 001 to enable 16-bit interface mode
- Set EERE = 1 to start E2PROM read operation
- Wait for 2 system clock cycles (two NOP instructions are executed)
- The data corresponding to the destination address is updated to the internal 16-bit register of the controller. The user can use EEAR [0] And EEDR read the specified byte of data or all data.

5. Program ***FLASH*** erase operation

- Check the EEPE bit and wait for the FLASH controller to be idle
- Set EEAR [14: 0] for the target page address to be erased, the program FLASH page size is 1K bytes, So EEAR [14:10] will be set to 0 as the page address, EEAR [9: 0]
- Set EEPM [3: 0] = 1X01, where EEPM [2] can be set to 0 or 1
- Set EEMPE = 1 while EEPE = 0
- In the four cycles, set EEPE = 1 to start the program FLASH erase process

6. Program ***FLASH*** programming operation

- Check the EEPE bit and wait for the FLASH controller to be idle
- Set the 16-bit programming data via EEAR [0] and EEDR
- Set EEAR [14: 1] as the destination address, where the address is 2 bytes
- Set EEPM [3: 0] = 1X10, where EEPM [2] can be set to 0 or 1
- Set EEMPE = 1 while EEPE = 0
- In the four cycles, set EEPE = 1 to start the FLASH programming flow

【important】

For the LGT8F328D, it is necessary to first enable the ECCR register by performing the E2PROM operation

E2PROM controller and configure the size of the E2PROM. The following operation is the same as LGT8F48D / 88D / 168D.

Page 24

General Purpose **I / O** Registers **- GPIOR2**

GPIOR2 - General Purpose **I / O** Register **2**

GPIOR2: 0x2B (0x4B)		Default: 0x00
GPIOR2		GPIOR2 [7: 0]
R / W		R / W
Initial value		0x00
Bit definition		
[7: 0]	GPIOR2	General purpose I / O register 2 for storing user-defined data

General Purpose **I / O** Register **- GPIOR1**

GPIOR1 - General Purpose **I / O** Register **1**

GPIOR1: 0x2A (0x4A)		Default: 0x00
GPIOR1		GPIOR1 [7: 0]
R / W		R / W
Initial value		0x00
Bit definition		
[7: 0]	GPIOR1	General purpose I / O register 1 for storing user-defined data

General Purpose **I / O** Register **- GPIOR0**

GPIOR0 - General Purpose **I / O** Register **0**

GPIOR0: 0x1E (0x3E)		Default: 0x00
GPIOR0		GPIOR0 [7: 0]
R / W		R / W
Initial value		0x00
Bit definition		
[7: 0]	GPIOR0	The general purpose I / O register 0 is used to store user-defined data

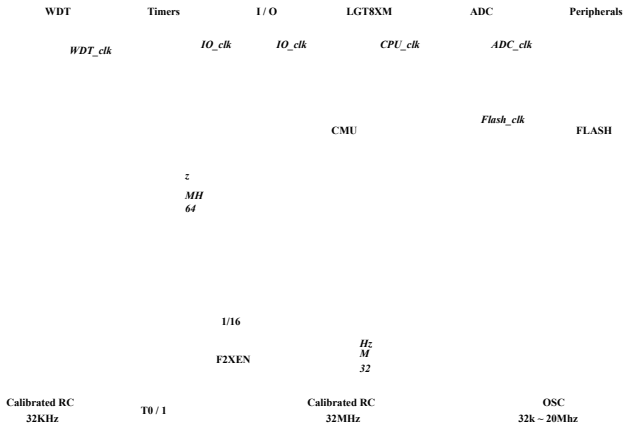
Page 25

System clock and configuration

System clock distribution

The LGT8FX8D supports multiple clock inputs. The system can work in three main clock sources, namely internal 32KHz Calibrated RC Oscillator, Internal 32MHz Calibrated RC Oscillator, and External 400KHz ~ 20MHz Crystal Input.

The following figure shows the LGT8FX8D clock system distribution, CMU is the center of the entire clock management, responsible for the system clock frequency, The same module to generate a separate clock and control the clock and so on. In general applications, do not all the clock with the same When working in order to reduce system power consumption, the system power management switches the unused module clock according to the different sleep modes. For details on the operation, refer to the section on power management.



CPU_clk

The CPU clock is used to drive the LGT8XM core and SRAM operation. Such as driving the general working register, the status sent Such as the deposit. After the CPU clock is stopped, the kernel will not continue to execute the instructions and perform the calculations.

IO_clk

The IO clock is used to drive most peripheral modules, such as timers / counters, SPI, USART, and so on. The IO clock is also used for flooding Active external interrupt module. When the IO clock is stopped due to hibernation, some can use the peripheral part of the wake-up system In standalone clock or asynchronous mode. For example, TWI's address recognition function can wake most of the sleep mode, at this time The address recognition part operates in asynchronous mode.

Flash_clk

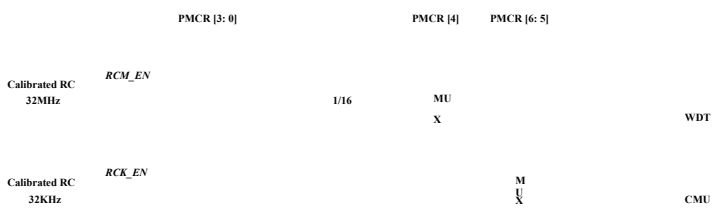
The FLASH clock is used to generate the FLASH interface access timing. The FLASH clock is homologous to the system clock. FLASH clock master To use the program through the FLASH controller on the program FLASH and data FLASH access.

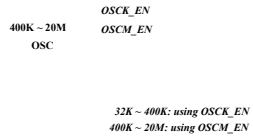
Asy_clk

Asynchronous timer clock. The timer / event counter can be driven directly using an external clock or crystal (32.768K). This independence Clock mode, the timer can remain running while the system is in sleep mode.

Clock source selection

LGT8FX8D support four kinds of clock source input, the user can PMCR register to achieve the clock source enable control As well as the completion of the main clock switch. The following is the control structure of the PMCR:





LGT8FX8D internal OSC oscillator can work in high frequency and low frequency two modes, the user needs according to the external crystal The actual size of the internal OSC oscillator is controlled in the correct mode. The same internal RC oscillator is also divided into high frequency and Low frequency two kinds. The lowest 4 bits of the PMCR register are used to control these four clock sources. The control relationship is as follows:

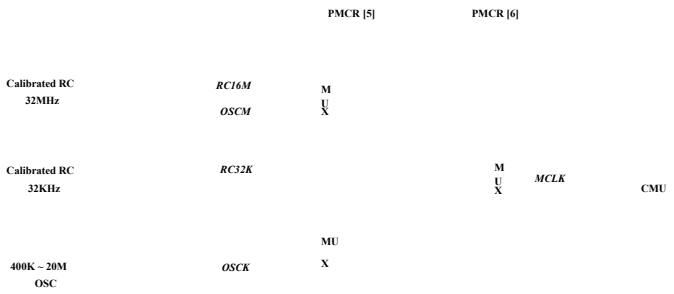
PMCR	Corresponds to the clock source
PMCR [0]	32MHz RC enable control, 1 enabled, 0 off
PMCR [1]	32KHz RC enable control, 1 enabled, 0 off
PMCR [2]	400K ~ 20MHz OSC mode enabled, 1 enabled, 0 closed
PMCR [3]	32K ~ 400K OSC mode enabled, 1 enabled, 0 off

LGT8FX8D system power, the default use of 32MHz RC as the system clock source, the kernel work in the clock source of 8 points Frequency (2MHz). The user can change the default configuration by setting the PMCR register and the system prescaler register (CLKPR).

If the user needs to change the main clock source configuration, it is necessary to ensure that the clock source after switching is stable before switching the clock. State. It is therefore necessary to enable the required clock source by PMCR [3: 0] before switching to the master clock source and wait for the clock to stabilize After the set to switch.

When the user switches the master clock to the external crystal, although the user enables the external crystal, but does not rule out the configuration error or Crystal failure can not cause crystal vibration. If you switch to an external crystal at this time, the system will stop working immediately after switching. because This, from the system reliability considerations, it is recommended by opening the watchdog timer, from the software design point of view to avoid such questions question.

After the clock source is enabled and waiting for stabilization, the master clock can be switched by PMCR [6: 5]. Where PMCR [5] is used to select Yes Internal RC oscillator and external crystal, PMCR [6] is used to select the high-speed clock source and low-speed clock source.



Main clock source selection:

PMCR [6]	PMCR [5]	Main clock source
0	0	Internal 32MHz RC Oscillator (System Default)
0	1	External 400K ~ 20MHz high-speed crystal
1	0	Internal 32KHz RC oscillator

1	1	External 32K ~ 400KHz low-speed crystal

Clock source control timing

In order to protect the PMCR register from accidental modification, the modification of the PMCR register requires strict installation of the specified timing Row. The highest bit of the PMCR register (PMCR [7]) is used to implement timing control. Before modifying the other bits of the PMCR, the user must The PMCR [7] must first be set and the value of the other registers of the PMCR should be changed during the six cycles after the set operation. 6 After the cycle, the direct modification to the PMCR will fail.

Below to switch to the external high-speed crystal, for example, list the recommended steps:

(1) Enable the clock source

- Set PMCR [7] = 1
- Set PMCR [2] = 1 in six cycles to enable external high-speed mode external crystal
- wait for external crystal stability (waiting time due to different crystal and different, generally us level can wait)

(2) toggle the main clock source

- Set PMCR [7] = 1
- Set PMCR [6: 5] = 01 in six cycles, the system will automatically switch the working clock to the external crystal
- Perform several NOP operations to improve stability (optional operation)

/ Note **/** : In the above operation to switch the main clock, to ensure that the current system clock to work properly, in the switch to the external crystal After the vibration, you can turn off the internal ***RC*** oscillator before .

System clock prescaler control

The LGT8FX8D has a system clock prescaler internally that can be controlled by the clock prescaler register (CLKPR). This function can be used to reduce system power consumption when the system does not require very high processing power. Prescaler settings for system support The clock source is valid. Clock prescaler can affect the kernel execution clock and so the synchronization peripherals.

When switching between different clock prescaler settings, the system clock prescale ensures that no hair is generated during the switching process Thorns, but will ensure that there will be no high frequency of the middle state. The frequency switching is performed immediately, and when the register change takes effect, Up to 2 to 3 current system clock cycles, the system clock switches to the new divide clock.

In order to avoid erroneous operation on the clock divider register, the modification to CLKPR must also follow a special timing flow Cheng:

- Set the clock prescaler change enable bit (CLKPCE) to 1, CLKPR other bit is 0
- Write the required value to CLKPS in four cycles, while CLKPCE is written to 0

Before changing the clock prescaler register, it is necessary to disable the interrupt function to ensure that the write timing can be performed intact. Refer to the Register Description section of this section for a detailed definition of the main clock prescaler register CLKPR.

【important】:

LGT8FX8D maximum operating frequency of 20MHz, so when the choice of internal 32MHz RC as the main clock source, be sure to Make sure CLKPR is set to the correct frequency division (minimum 2).

Internal RC oscillator calibration

The LGT8FX8D contains two calibratable RC oscillators inside, and can be calibrated to within ± 1% accuracy. its The 32MHz RC is used by default for the system operating clock.

Each LGT8FX8D is manufactured before the internal 32MHz RC has been calibrated, and the calibration value written to the system with

Set the information area. The user's client accesses this calibration value through the RCCAL register of the I / O register space.
The internal 32KHz RC is not calibrated before production, and if the user needs a very accurate low frequency clock,

You can calibrate the RC oscillator, and the calibration value written to the LGT8FX8D comes with the data FLASH. At each time

After the chip is started, the calibration value is read by software and written to the RCKCAL register to complete the 32KHz RC oscillator

Of the calibration. Please refer to the following description of the specific calibration method.

Internal ***32KHz RC*** calibration method:

Before calibrating the internal 32KHz RC, it is necessary to be able to measure the current internal 32KHz clock frequency. Relatively simple side

The method is to switch the system clock to the internal 32KHz RC oscillator. And then output a relatively easy to measure by I / O

Square wave The frequency of the internal RC is obtained by measuring the square wave frequency.

Above is the RC calibration method, 32KHz RC oscillator calibration method and the same. After getting RCKCAL, you can

To write its value to a reserved area of the data FLASH (E2PROM) for subsequent software calibration.

Register definition

32MHz RC Oscillator Calibration Register - RCCAL

RCCAL - 32MHz RC Calibration Register

RCCAL: 0x66 Default: 0x00

RCCAL	RCCAL [7: 0]
R / W	R / W
Initial value	0x00

Bit definition

[5: 0]	RCCAL	6bit RC calibration value, the system will power up, the value of the register will be the system configuration information in the RC calibration Value replacement.
[7: 6]	-	Keep not used

Clock Source Management Register - PMCR

PMCR - clock source management register

PMCR: 0xF2 Default: 0x01

PMCR	PMCE	CLKFS / CLKSS	WCLKS	OSCKEN	OSCMEN	RCKEN	RCMEN
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	0	0/0	0	0	0	0	1

Bit definition

[0]	RCMEN	Internal 32MHz RC oscillator enable control, 1 enabled, 0 disabled
[1]	RCKEN	Internal 32KHz RC oscillator enable control, 1 enabled, 0 disabled
[2]	OSCMEN	External high frequency crystal enable control, 1 enabled, 0 disabled
[3]	OSCKEN	External low frequency crystal enable control, 1 enabled, 0 disabled
[4]	WCLKS	WDT clock source selection, 0 - select internal 1MHz / RC; 1 - internal 32KHz
[5]	CLKSS	Main clock source selection control, select the clock source type, refer to the clock source selection section
[6]	CLKFS	Main clock source frequency control, select the clock frequency type, refer to the clock source selection section
[7]	PMCE	The PMCR register changes the enable control bits. Before changing other locations of the PMCR, you must first set this bit and then in four cycles Set the value of the other bits.

Master clock prescaler register - **CLKPR**

CLKPR - Master clock prescaler register

CLKPR: 0x61 Default: 0x03

CLKPR	CLKPCE	CLKOEN1	CLKOEN0	-	CLKPS3	CLKPS2	CLKPS1	CLKPS0
R / W	R / W	R / W	R / W	-	R / W	R / W	R / W	R / W
Initial value	0	0	0	-	0	0	1	1

Page 30

Bit definition

		Clock prescaler selection bit				
		CLKPS3	CLKPS2	CLKPS1	CLKPS0	Frequency division parameters
[3: 0]	CLKPS	0	0	0	0	1
		0	0	0	1	2
		0	0	1	0	4
		0	***0***	***1***	***1***	***8 (*** default configuration ***)***
		0	1	0	0	16
		0	1	0	1	32
		0	1	1	0	64
		0	1	1	1	128
		1	0	0	0	256
		Other values				Keep it
[4]	-	Keep not used				
[5]	CLKOEN0	Set whether the system clock is output on the PB0 pin				
[6]	CLKOEN1	Set whether the system clock is output on the PE5 pin				
[7]	CLKPCE	Clock prescaler changes clock control Before changing the other bits of the CLKPR register, you must first set CLKPCE to 1, And then set the other bits in the next four system cycles. After the four cycles, CLKPCE is automatically cleared.				

32KHz RC Oscillator Calibration Register - **RCKCAL**

RCKCAL - 32MHz RC Calibration Register

RCKCAL: 0x67 Default: 0x00

RCKCAL	RCKCAL [7: 0]
R / W	R / W
Initial value	0x00

Bit definition

[5: 0]	RCKCAL	6bit RC calibration value for writing calibration values to the RCKCAL register for 32 kHz RC oscillation Calibration of the device. Refer to the RC calibration section of this chapter for specific calibration methods.
[7: 6]	-	Keep not used

Power management

Overview

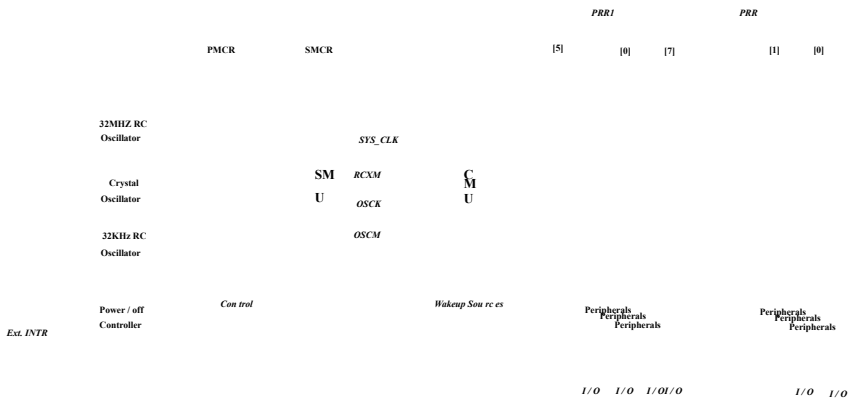
Sleep mode is used to turn off unused modules in the MCU, thereby reducing system power consumption. LGT8FX8D offers non-
Often flexible and diverse sleep mode and module controller, the user can be based on the application, to achieve the best low-power configuration.

When the low voltage detection (LVD) module is enabled, the LVD will continue to operate in Sleep mode. In order to further reduce the system
Power consumption, you can enter the sleep mode before you turn off.

LGT8FX8D support power-down mode, power-down mode, the system's digital part, most of the I / O and analog modules will
In a power-down state. Power-down mode minimizes system power consumption. Power-down mode can only be woken up by the specified external pin
The process is consistent with the system power-up process. The software can get the status of the system by reading the MCU status register.

The LGT8FX8D internally includes a precision calibrable 32KHz RC oscillator that allows the user to
Task, the system clock is switched to 32KHz RC, using the appropriate sleep mode. This can also be achieved in non-power-down mode
The ideal system power consumption. If the user external 32768Hz crystal, you can also switch the main clock to the external crystal, then
After the closure of the unused clock source and other analog modules, and then enter the sleep mode, so you can further save power
Consumption.

System power management diagram:



As shown in the figure above, the LGT8FX8D controls the entire system through the Sleep Mode Controller (SMU) and the Clock Management Unit (CMU)
Power consumption of the system. From the level of power savings, we can divide the power consumption into three levels, the first level is sent through the PRR
The register control module operates the clock by turning off the clock without using the module, saving the system to run the dynamic power consumption. General situation
Circumstances, this level can save power consumption is not obvious. The second level is by switching the main clock source to the low frequency clock and turning off
Closed with no use of the clock source module and other analog modules, this model can basically get very impressive system running
Power consumption and hibernation power consumption. The third level is through the system into the power (Power / off) mode, power off mode LGT8FX8D
The operating power is minimal, and this mode can only be woken up by an external interrupt pin. Wake up from power down mode, the software can pass
The MCUSR register reads the state before reset.

Sleep mode

LGT8FX8D supports five sleep modes, the user can choose the appropriate sleep mode according to application requirements. SMCR deposit Device contains the sleep mode control settings, the implementation of SLEEP instructions, the kernel into sleep mode. To get more ideal Sleep power, it is recommended that the kernel into the sleep mode before the closure of all unused clock and analog modules. But need to note It is intended that some wake-up sources need to work clocks, and if you need to use such wake-up sources, keep the relevant clock source Working state.

Sleep mode and wake mode:

	Valid clock				Wake source						
Sleep Mode	FLASH_CLK CPU_CLK	IO_CLK	ADC_CLK	ASY_CLK	PIN Change INT1	INT0	TWI Address Match	Timer2	ADC	WDT	Other INT	Other I / O
Idle	XXXX				X	X	XXXX					X
ADC Noise Reduction		XX			X	X	XXX					
Power / Down					X	X	X			X		
Power / Off S0 (With RC32K)					X					X		
Power / Off S1 (Without RC32K)					X							

If you need to enter the above five kinds of sleep mode, SMCR in the SE bit must be set to enable sleep mode control. then Execute a SLEEP instruction. SM0 / 1/2 in SMCR is used to select a different sleep mode. Please refer to the specific information Test the following description.

When the MCU is in Sleep mode, if the wake source is active, the MCU will wake up after 4 cycles and continue execution instruction. If the interrupt remains active, the interrupt will also respond immediately to the interrupt service routine. If in SLEEP mode A system reset occurs and the MCU will be woken up and executed from the reset vector.

When the MCU is in Power / Off mode, the system can wake up via external interrupt INT0 / 1, wake up after the MCU will Continue from the previous position before sleep.

IDLE mode

When SM2 ... 0 is set to 000, after the SLEEP instruction is executed, the MCU enters IDLE mode and IDLE mode will be turned off Out of the kernel working clock, in addition to other peripherals can work properly.

IDLE mode can be awakened by external interrupts and internal interrupts. If you do not need to use the comparator and ADC To wake up the source, it is recommended to turn it off.

IDLE mode because only the kernel to run off the clock, it can not be significantly reduced power consumption. IDLE mode , The kernel will also stop executing and fetch instructions, thus reducing the power consumption of the internal program FLASH.

But IDLE mode has a more flexible way to wake up, the user can reduce the system by reducing the main clock and do not need Of the module to get more ideal running power consumption.

ADC noise suppression mode

When SM2 ... 0 is set to 001, the MCU enters the ADC noise suppression mode after executing the SLEEP instruction. In this mode, The kernel and most peripherals will stop working, ADC, external interrupt, TWI address match, WDT, and work on asynchronous Clock mode 2 / timer 2 can work normally.

ADC noise always used to provide a good working environment for ADC conversion. Reduce the number of modules Quasi-converted high frequency interference. After entering this mode, the ADC will automatically start the sample conversion, the converted data is saved to the ADC number

After the register, the ADC conversion end interrupt will wake the MCU from ADC noise mode.

Power / Down mode

When SM2 ... 0 is set to 010, the MCU enters Power / down mode after executing the SLEEP instruction. This pattern
, The system will turn off the working clock of all modules. This mode can only be used because the working clock of all modules is turned off
Wake-up via asynchronous mode, external interrupt, TWI address match, and WDT operating in independent clock source mode
Generates a wake-up signal in this mode.

This mode can turn off all modules except the master clock source. In order to achieve a more ideal operating power, it is recommended to enter
Before entering this mode, switch the system master clock to internal 32K RC or external 32KHz low frequency crystal, and then turn off the
To the unused clock source and the analog module.

Power / Off S0 mode

When SM ... 0 is set to 110, the MCU will enter Power / Off S0 mode after executing the SLEEP instruction. enter
After power / Off S0, other clock sources are turned off except for internal 32KHz RC. This mode can be externally interrupted
INT0 / 1 wake-up; if the WDT interrupt function is enabled, it can also be used to wake up the WDT.

Power / Off S1 mode

When SM ... 0 is set to 011, the MCU will enter Power / Off S1 mode after executing the SLEEP instruction. enter
Power / Off S0, all system clocks are turned off. This mode can only be woken up by an external interrupt INT0 / 1.

Before entering Power / Off mode, you need to enable the wake-up source in advance and set the appropriate wake-up condition; for external interrupt
Wake up, the user needs to set the interrupt enable and configure the trigger level according to the application needs.

FLASH power control and fast wakeup

When the system is in SLEEP mode, the kernel will not continue to execute the instruction, then you can choose to turn off the FLASH power
Source to achieve lower standby power consumption. This function can be implemented by the FPDEN bit control of the MCUCR register;

In Power / Off mode, the system can use an external interrupt or WDT wake-up, in order to filter out the external signal may
Of the interference, the internal wake-up circuit contains a configurable filter circuit, the user can select the appropriate filter width
degree. The configuration of the filter circuit can be implemented by the FWKPEN of the MCUCR register.

MCUCR [FWKPEN] Filter width control:

FWKPEN	Filter width
0	260us (default)
1	32us

Register description

Sleep mode control register - SMCR

SMCR - Sleep mode control register

SMCR: 0x33 (0x53) Default: 0x00

SMCR			SM2	SM1	SM0	SE
R / W		-	R / W	R / W	R / W	R / W
Initial value		-	0	0	0	0

Bit definition

[0]	SE	Sleep mode enable control bit, set to 1, the implementation of SLEEP instruction, the kernel will enter the sleep What mode. The SE bit protects the system from accidentally entering sleep mode. It is recommended that the user set this bit After 1, followed by the SLEEP instruction. After waking up, it is recommended to clear the SE bit immediately.

		Sleep mode selection SM2	SM1	SM0	Mode description
		0	0	0	IDLE mode
		0	0	1	ADC noise suppression mode
[3: 1]	SM	0	1	0	Power / Down mode
		0	1	1	Power / Off S1 mode
		1	1	0	Power / Off S0 mode
		1	0	0	Power / Off Lock
		Others			Keep not used
[7: 4]	-	Keep not used			

Power Saving Control Register - **PRR**

PRR - Power saving control register

PRR: 0x64 Default: 0x00

PRR	PRTWI	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	PRUSART0	PRADC
R / W	R / W	R / W	R / W	-	R / W	R / W	R / W	R / W
Initial value	0	0	0	-	0	0	0	0

Bit definition

[0]	PRADC	Set to 1 to turn off the ADC controller clock
[1]	PRUSART0	Set to 1 to turn off the clock of the USART0 module
[2]	PRSPI	Set to 1 to turn off the SPI module clock
[3]	PRTIM1	Set to 1 to turn off the Timer of Timer / Event Counter 1
-	-	Keep not used
[5]	PRTIM0	Set to 1 to turn off the Timer of Timer / Event Counter 0
[6]	PRTIM2	Set to 1 to turn off the Timer of Timer / Event Counter 2
[7]	PRTWI	Set to 1 to turn off the TWI module clock

Power Saving Control Register - **PRR1**

PRR1 - Power saving control register 1

PRR1: 0x65 Default: 0x00

PRR1		PRWDT	-	-	PREFL	PRPCI	-
R / W		R / W	-	-	R / W	R / W	-
Initial value		0	-	-	0	0	-

Bit definition

[0]	-	Keep not used
[1]	PRPCI	Set to 1, turn off external pin change and external interrupt module operating clock
[2]	PREFL	Set to 1, turn off the FLASH controller interface timing clock
[4: 3]	-	Keep not used
[5]	PRWDT	Set to 1 to turn off the WDT counter clock
[7: 6]	-	Keep not used

MCU Control Register - **MCUCR**

MCUCR - **MCU** control register

MCUCR: 0x35 (0x55) Default: 0x00

MCUCR	FWKEN	FPDEN	PUD	-	-	-	IVSEL	IVCE

R / W	R / W	R / W	R / W	-	-	-	R / W	R / W
Initial value	0	0	0	-	-	-	0	0

Bit definition

[0]	IVCE	Interrupt vector selection change enable bit, before changing IVSEL, you need to set this bit first After setting the IVSEL in 6 cycles.
[1]	IVSEL	Interrupt vector selection bit, this bit is set after the interrupt vector address will be based on the IVBASE register The value is mapped to the new address. For detailed mapping addresses, refer to the IVBASE register description
[2]	-	Keep not used
[3]	-	Keep not used
[4]	PUD	Global pull-up disabled bit
[5]	-	Keep not used
[6]	FPDEN	Flash Power / down enable control 0: After the system SLEEP, the FLASH remains powered on 1: System SLEEP after the FLASH power
[7]	FWKEN	Fast wake-up mode enable control, only valid for Power / Off mode 0: 260us filter delay 1: 32us filter delay

System control and reset

Overview

After the system is reset, all I / O registers are set to their initial values, starting with the reset vector
carried out. On the interrupt vector address of the LGT8FX8D, you must jump to the reset process with an RJMP - relative jump instruction sequence. If the program is not used to interrupt, no interrupt source is enabled, the interrupt vector will not be used, the interrupt vector area It can be used to store the user's program code.

When the reset is active, all I / O ports immediately enter their initial state. Most I / O initialization states are input
And turn off the internal pull-up resistor. I / O with analog input function is also initialized to digital I / O function.

When the reset becomes inactive, the internal timer counter of the LGT8FX8D starts to be used for widening the reset. Widen the reset The width of the signal is used to ensure that the power supply in the system and the clock and other modules enter a stable state.

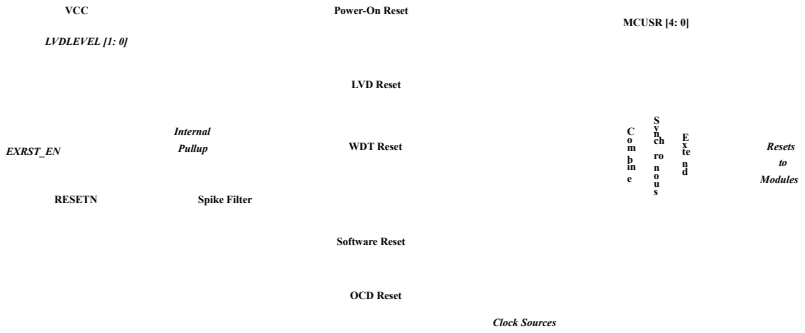
Reset source

LGT8FX8D supports a total of six reset sources:

- Power-on reset: When the operating voltage of the system is low, the reset value of the internal POR module is valid.
- External reset: When a pulse above the minimum reset width appears on the chip's RESETN pin, the external Reset is valid.
- Watchdog reset: When the watchdog module is enabled, the system will reset if the watchdog timer times out.
- Low voltage reset: LGT8FX8D has a low voltage detection module (LVD) inside, when the system power supply is lower than LVD When the reset threshold is set, the MCU will also be reset.
- Software reset: LGT8FX8D internal has a dedicated software to trigger the reset register, the user can through this The register resets the MCU at any time.
- OCD Reset: OCD Reset is issued by the debugger module for direct resetting the MCU core.

Reset system structure diagram:

Reset Flags



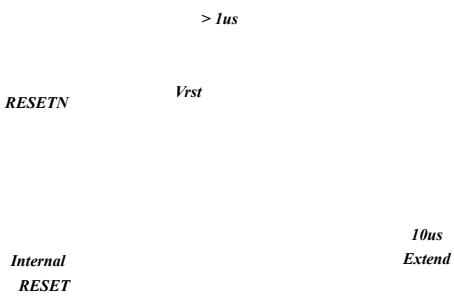
Power on reset

The power-on reset signal is generated by an internal voltage detection circuit. When the system power supply (VCC) is lower than the detection threshold, power up The reset signal is valid. Refer to the Electrical Parameters section for the detection threshold for power-on reset.

Power-on reset circuit to ensure that the chip in the power process in a reset state, the chip after power from a known Steady state to start running. Power-on reset signal will be the chip internal expansion of the counter to ensure that after the power of the internal An analog module, such as an RC oscillator, can enter a stable operating state.

External reset

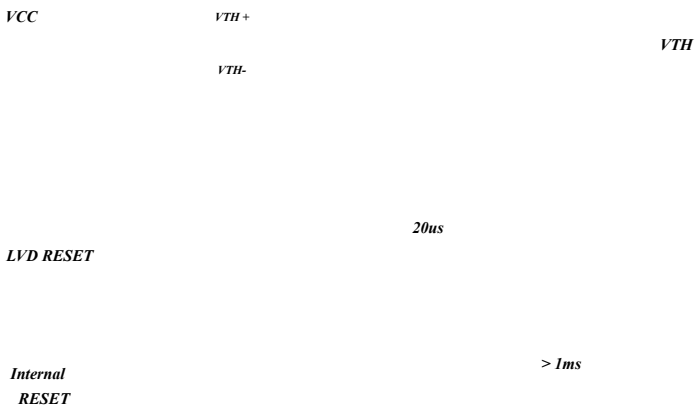
A low level is applied to the external reset pin (RSTN), and the external reset is active immediately. The width of the low level is greater than one Minimum reset pulse width requirement. External reset is asynchronous reset, even if the chip does not have clock operation, external reset is still possible Enough to reset the chip. The external reset pin of the LGT8FX8D can also be used as a general purpose I / O. Power on the chip Later, the default is as an external reset function. The user can switch through the register configuration, close the pin of the external reset function, Which can be used as a normal I / O. For details, refer to the description section of the IOCR register.



Low voltage detection **(LVD)** reset

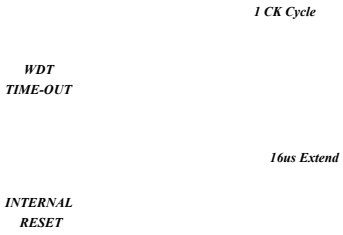
The LGT8FX8D internally contains a programmable low voltage detection (LVD) circuit. LVD is also the detection of VCC voltage changes,
However, unlike the power-on reset, the LVD can select the threshold for the detection voltage. The user can either through the system configuration bits or directly
The VDTCR register is used to select between three different voltage thresholds. LVD voltage detection circuit with ± 10mV ~
± 50mV hysteresis, used to filter out the VCC voltage jitter. When LVD is enabled, if the VCC voltage drops to
Set the reset threshold, LVD reset will be active immediately. When VCC is increased above the reset threshold, the internal reset is energized

The path starts, and the reset continues to widen for at least 1 millisecond.



Watchdog reset

When the watchdog timer overflows, the watchdog system reset function is enabled and will immediately generate a cycle of the system
Reset signal. The watchdog reset signal is also broadened by the internal delay counter. Watchdog controller for detailed operation,
Please refer to the detailed section below.



Software reset, **OCD** reset

Software reset is the user through the operation of the VDTCR register sixth trigger, software reset timing and watchdog reset
Completely similar. The interior will reset the signal by 16us.

OCD reset is generated by the internal debugger unit, OCD reset is generally controlled by the debugger, the user software
Method triggers OCD reset.

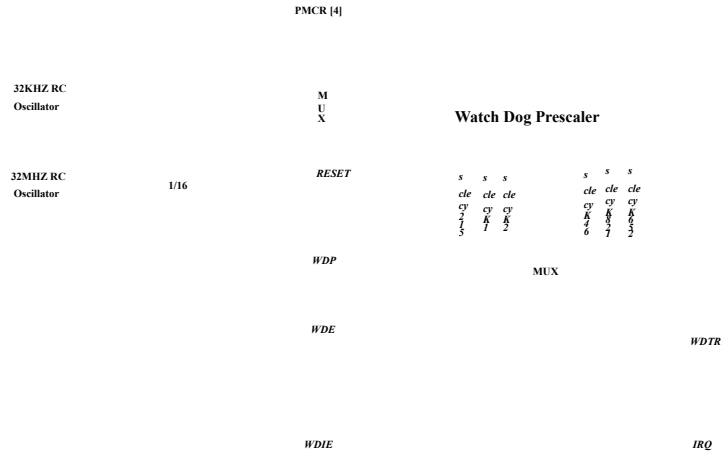
Pin multiplexing control

Watchdog timer

- Clock optional internal 32KHz RC or internal 32MHz RC 16-way (2MHz)
- Supports interrupt mode, reset mode and reset interrupt mode
- The timer can be up to 8 seconds

The LGT8FX8D contains an enhanced watchdog timer (WDT) module internally. The WDT timer can operate clock
Is the internal 32KHz RC oscillator, it can also be internal 32MHz RC oscillator 16 frequency. WDT counter overflow
After the output, you can output an interrupt or a system reset signal. In normal use, the need for software to implement one
WDR - The watchdog timer reset instruction restarts the counter before overflowing. If the system does not have to execute WDR
Instruction, the WDT will generate an interrupt or a system reset.

The structure of the watchdog timer is shown in the following figure:



In interrupt mode, an interrupt request signal is generated after a WDT overflow. You can use this interrupt as a dormant mode
Type wake-up signal can also be used as a general system timer. For example, you can use this interrupt to limit a
The execution time of an operation terminates a current task in the overflow. In system reset mode, the WDT is counting
A system reset signal is generated immediately after overflow. The most typical use is to prevent the system from crashing or running away. First
Three modes, is to reset the interrupt mode, combined with the interrupt and reset the two functions. First the system will respond to the WDT
Off function, exit WDT interrupt reset program immediately after switching to reset mode. This function can be supported at reset
Before the preservation of some of the more critical parameter information.

In order to prevent the WDT from being accidentally disabled, the operation to turn off the WDT must be performed at a strictly defined timing. the following
The code describes how to turn off the watchdog timer. The following example assumes that the interrupt has been disabled, so the entire operation flow
The process will not be interrupted.

Assembly code

```
WDT_OFF:
```

```
; Turn off global interrupt
CLI
; Reset watchdog timer
WDR
; Clear WDRF in MCUSR
IN r16, MCUSR
ANDI r16, ~ (1 << WDRF)
OUT MCUSR, r16
; Write logical one to WDCE and WDE
; Keep old Prescaler setting to prevent unintentional time-
out
LDS r16, WDTCSR
ORI r16, (1 << WDCE) | (1 << WDE)
STS WDTCSR, r16
; Turn off WDT
LDI r16, (0 << WDE)
STS WDTCSR, r16
; Turn on global interrupt
SEI
RET
```

C language code

```
void WDT_OFF (void)
{
  __disable_interrupt ();
  __watchdog_reset ();
  / * Clear WDRF in MCUSR * /
  MCUSR & = ~ (1 << WDRF);
  / * Write logical one to WDCE and WDE * /
  / * Keep old Prescaler setting to prevent unintentional time-
  out * /
  WDTCSR | = (1 << WDCE) | (1 << WDE);
  / * Turn off WDT * /
  WDTCSR = 0x00;
  __enable_interrupt ();
}
```

[Tips]

If the WDT is accidentally enabled, such as the program running, the chip will be reset, but the WDT is still enabled. If used

The WDT is not processed in the user code, which will cause a cyclic reset. To avoid this, it is recommended that the user software initialize the program

Clear the watchdog reset flag (WDRF) and the WDE control bit.

The following code describes how to change the watchdog timer timeout value.

Assembly code

```
WDT_TOV_Change:
  ; Turn off global interrupt
  CLI
  ; Reset watchdog timer
  WDR
  ; Start timed sequence
  LDS r16, WDTCSR
  ORI r16, (1 << WDCE) | (1 << WDE)
```

```
    STS WDTCSR, r16
    ; - Got for cycles to set the new value from here -
    ; Set new time-out value = 64k cycles
    LDI r16, (1 << WDE) | (1 << WDP2) | (1 << WDP0)
    STS WDTCSR, r16
    ; - Finished setting new value, used 2 cycles -
    ; Turn on global interrupt
    SEI
    RET
```

C language code

```
void WDT_TOV_Change (void)
{
    __disable_interrupt ();
    __watchdog_reset ();
    / * Start timed sequence * /
    WDTCSR | = (1 << WDCE) | (1 << WDE);
    / * Set new time-out value = 64K cycles * /
    WDTCSR | = (1 << WDE) | (1 << WDP2) | (1 << WDP0);
    __enable_interrupt ();
}
```

【Instructions for use】

It is recommended to reset the watchdog timer before changing the WDP configuration bits. Because it is possible to change the WDP bit to a relatively small time-out period Can cause the watchdog to reset on time.

Register definition

Low Voltage Detection **(LVD)** Control Register **- VDTCR**

VDTCR - LVD control register

VDTCR: 0x62				Default: 0x00 or loaded from the system configuration information			
GPIOR2	CE	SWR	-	VDTS1	VDTS0	LVREN	VDTEN
R / W	R / W	W	-	R / W	R / W	R / W	R / W
Initial value			0x00 or loaded from the system configuration information				

Bit definition

[0]	VDTEN	Low voltage detection module enable control, 1 enabled, 0 disabled
[1]	LVREN	Low-voltage reset function enable control, 1 enabled, 0 disabled
[3: 2]	VDTS	Low voltage detection threshold configuration bit VDTS = 00: 1.8V VDTS = 01: 2.7V VDTS = 10: 4.3V
[5: 4]	-	Keep not used
[6]	SWR	Soft reset enable bit, this bit is cleared to generate a software reset
[7]	CE	VDTCR value change enable bit Before changing the value of the VDTCR register, the user must first write this bit to 1, after which

Change the value of other bits of VDTCR in 4 clock cycles. After four cycles CE is automatically cleared,
The update operation to the VDTCR register is invalid.

IO Special Power Control Register - IOCR

IOCR - IO Special Function Control Register

IOCR: 0xF0 Default: 0x00

IOCR	CE	STOSC1	STOSC0	DACEN1	DACEN0	XIEN	RVIO_EN	EXIO_EN
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	0x00 or loaded from the system configuration information							

Bit definition

0	EXIO_EN	PC6 pin defaults to the reset function, setting this bit to 1 will disable the external reset function, reset Function disabled, PC6 can be used as a normal I / O
1	RVIO_EN	VREF pin defaults to analog input function, set this bit to 1, will turn off the analog input function, This pin can be used as PE6
2	XIEN	External clock input enable control
3	DACEN0	DAO output enable
4	DACEN1	DAO1 output enable
5	STOSC0	Low speed crystal start control
6	STOSC1	High speed crystal start control
7	CE	IOCR value changes enable bit Before changing the value of the IOCR register, the user must first write this bit to 1, followed by 4 During the clock cycle, change the value of the IOCR other bits. After four cycles CE is automatically cleared, yes IOCR register update operation is invalid.

MCU Status Register - MCUSR

MCUSR - IO special function control register

MCUSR: 0x34 (0x54) Default: 0x00

MCUSR	SWDD	PDRF	-	OCDRF	WDRF	BORF	EXTRF	PORF
R / W	R / W	R / W	-	R / W	R / W	R / W	R / W	R / W
Initial value	0x00							

Bit definition

[0]	PORF	Power on reset flag, write 0 clear
[1]	EXTRF	External reset flag, power-on reset is automatically cleared, or write 0 cleared
[2]	BORF	Low voltage detection reset, power-on reset automatically cleared, or write 0 clear
[3]	WDRF	Watchdog reset flag, power-on reset automatically cleared, or write 0 clear
[4]	OCDRF	OCD debugger reset flag, power-on reset is automatically cleared, or write 0 cleared
[5]	PDRF	Wake up from the Power / off mode. Refer to the Power Management section for details.
[6]	-	Keep not used
[7]	SWDD	SWD interface disable bit. Writing a 1 closes the SWD interface. After the SWD interface is shut down, debugging and ISP operations can not be performed. If the user program is closed SWD interface, you can switch through the process of power down the way to prohibit the internal program shipped Line, and then debug and ISP operation. SWD interface is closed, SWD occupied two I / O The interface can be used as a general purpose I / O. To avoid misuse of SWDD, the user needs to update the SWDD bit after the first four Write SWDD in one cycle to take effect.

[Tips]:

In order to use the reset flag information more accurately and effectively, it is advisable to read the reset flag as soon as possible before the program is initialized
Clear it.

Watchdog Control Status Register - **WDTCSR**

WDTCSR - WDT control and status register

Address: 0x60 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	WDIF	WDIE	WDP3	WDTOE	WDE	WDP2	WDP1	WDP0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
[7]	WDIF	WDT interrupt flag. The WDIF bit is set when the WDT is operating in interrupt mode and overflow occurs. When the WDT interrupt is enabled WDIE is set to "1" and the global interrupt is set, the WDT interrupt is generated. This bit is cleared when a WDT interrupt is executed WDIF bit, which can also be cleared by writing "1" to the WDIF bit.

Page 44

Bit	Name	description
[6]	WDIE	WDT interrupt enable control bit. WDT interrupt is enabled when the WDIE bit is set to "1" and the global interrupt is set. When the WDIE bit is set to "0", the WDT interrupt is disabled. The WDIE bit together with the WDE bit determines the watchdog's operating mode, as shown in the following table.

WDE	WDIE	mode	After overflow action
0	0	stop	no
0	1	Interrupt mode	Interrupted
1	0	Reset mode	Reset
1	1	Interrupt and reset mode	Reset after reset

Bit	Name	description
[5]	WDP3	WDT prescaler factor selection control 3rd bit. WDP [3] and WDP [2: 0] make up the WDT prescaler factor selection bit WDP [3: 0] to set the WDT Of the overflow period.
[4]	WDTOE	WDT off enable control bit. When the WDE bit is cleared, the WDTOE bit is set, otherwise the WDT will not be turned off. When WDTOE Bit is set, the hardware clears the WDTOE bit after 4 clock cycles.
[3]	WDE	WDT enable control bit. WDT is enabled when the WDE bit is set to "1". When the WDE bit is set to "0", the WDT is Forbidden. WDE can only be cleared if the WDTOE bit is set. To turn off the WDT that is already enabled, Must follow the following timing: 1. Set the WDTOE and WDE bits simultaneously, even if WDE has been set, the start of the shutdown operation You must also write "1" to the WDE bit before. 2. Write "0" to the WDE bit for the next four clock cycles. This will turn off the WDT. When the WDE bit is set to "1" and the WDT overflow is reset, the WDT reset system flag is set after resetting the system flag WDRF (Located in the MCUSR register). The WDE bit is set when the WDRF bit is set. therefore To clear the WDE bit, the WDRF bit must be cleared first.
[2: 0]	WDP	WDT prescaler factor selection control. Used to set the WDT overflow period. It is recommended to change the value of WDP when the WDT is not counting Changes in the value of WDP during the process will produce unpredictable WDT overflow.

Watchdog Prescaler Selection List:

WDP3	WDP2	WDP1	WDP0	Watchdog timer Number of overflow cycles	32KHz clock	2MHz clock
0	0	0	0	2K cycles	64ms	1ms

0	0	0	1	4K cycles	128ms	2ms
0	0	1	0	8K cycles	256ms	4ms
0	0	1	1	16K cycles	512ms	8ms
0	1	0	0	32K cycles	1s	16ms
0	1	0	1	64K cycles	2s	32ms
0	1	1	0	128K cycles	4s	64ms
0	1	1	1	256K cycles	8s	128ms
1	0	0	0	512K cycles	16s	256ms

Page 45

1	0	0	1	1024K cycles	32s	512ms
1	0	1	0			
1	0	1	1			
1	1	0	0			
1	1	0	1		Keep not used	
1	1	1	0			
1	1	1	1			

Interrupt and interrupt vector

- 28 interrupt sources
- Programmable vector start address

LGT8F48D / 88D / 168D / 328D interrupt resources are basically the same, the main difference is: LGT8F48D / 88D The interrupt vector is 1 instruction word (16 bits), while the LGT8F168D / 328D interrupt vector is 2 instruction words.

LGT8F48D / 88D interrupt vector list

LGT8F48D / 88D interrupt vector list:

Numbering	Vector address	Interrupt source signal	Interrupt source description
1	0x0000	RESET	External reset, power-on reset, watchdog reset, SWD debug reset, low voltage reset
2	0x0001	INT0	External interrupt request 0
3	0x0002	INT1	External interrupt request
4	0x0003	PCI0	Pin level interrupt 0
5	0x0004	PCI1	Pin level interrupt 1
6	0x0005	PCI2	Pin level interrupt 2
7	0x0006	WDT	Watchdog overflow interrupt
8	0x0007	TC2 COMPA	Timer 2 compare match A interrupt
9	0x0008	TC2 COMPB	Timer 2 compare match B interrupt
10	0x0009	TC2 OVF	Timer 2 overflow interrupt
11	0x000A	TC1 CAPT	Timer 1 input capture interrupt
12	0x000B	TC1 COMPA	Timer 1 compare match A interrupt
13	0x000C	TC1 COMPB	Timer 1 compare match B interrupt
14	0x000D	TC1 OVF	Timer 1 overflow interrupt
15	0x000E	TC0 COMPA	Timer 0 compare match A interrupt
16	0x000F	TC0 COMPB	Timer 0 compare match B interrupt
17	0x0010	TC0 OVF	Timer 0 overflow interrupt
18	0x0011	SPI STC	SPI serial transmission ends interrupt
19	0x0012	USART RXC	The USART receives an end interrupt
20	0x0013	USART UDRE	The USART data register is empty
twenty one	0x0014	USART TXC	The USART sends an end interrupt
twenty two	0x0015	ADC	ADC conversion end interrupt
twenty three	0x0016	EE_RDY	EEPROM ready to interrupt
twenty four	0x0017	ANA_COMP	Analog Comparator 0 interrupt
25	0x0018	TWI	Two-wire serial interface is interrupted
26	0x0019	ANA_COMP1	Analog Comparator 1 is interrupted
27	0x001A	-	Keep it
28	0x001B	PCI3	Pin level interrupt 3
29	0x001C	OPA0_COMP	OPA0 built-in timer compare match interrupt
30	0x001D	OPA1_COMP	OPA1 built-in timer compare match interrupt

LGT8F168D / 328D interrupt vector list

LGT8F168D / 328D Interrupt Vector List:

Numbering	Vector address	Interrupt source signal	Interrupt source description
1	0x0000	RESET	External reset, power-on reset, watchdog reset, SWD debug reset, low voltage reset
2	0x0002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request
4	0x0006	PCI0	Pin level interrupt 0
5	0x0008	PCI1	Pin level interrupt 1
6	0x000A	PCI2	Pin level interrupt 2
7	0x000C	WDT	Watchdog overflow interrupt
8	0x000E	TC2 COMPA	Timer 2 compare match A interrupt
9	0x0010	TC2 COMPB	Timer 2 compare match B interrupt
10	0x0012	TC2 OVF	Timer 2 overflow interrupt
11	0x0014	TC1 CAPT	Timer 1 input capture interrupt
12	0x0016	TC1 COMPA	Timer 1 compare match A interrupt
13	0x0018	TC1 COMPB	Timer 1 compare match B interrupt
14	0x001A	TC1 OVF	Timer 1 overflow interrupt
15	0x001C	TC0 COMPA	Timer 0 compare match A interrupt
16	0x001E	TC0 COMPB	Timer 0 compare match B interrupt
17	0x0020	TC0 OVF	Timer 0 overflow interrupt
18	0x0022	SPI STC	SPI serial transmission ends interrupt
19	0x0024	USART RXC	The USART receives an end interrupt
20	0x0026	USART UDRE	The USART data register is empty
twenty one	0x0028	USART TXC	The USART sends an end interrupt
twenty two	0x002A	ADC	ADC conversion end interrupt
twenty three	0x002C	EE_RDY	EEPROM ready to interrupt
twenty four	0x002E	ANA_COMP	Analog comparator interrupt
25	0x0030	TWI	Two-wire serial interface is interrupted
26	0x0032	ANA_COMP1	Analog Comparator 1 is interrupted
27	0x0034	-	Keep it
28	0x0036	PCI3	Pin level interrupt 3
29	0x0038	OPA0_COMP	Op amp 0 Built-in timer match interrupt
30	0x003A	OPA1_COMP	Op amp 1 built-in timer match interrupt

The reset vector for LGT8FX8D is executed from address 0x0000. In addition to the reset vector, other vector addresses are available The IVSEL and IVBASE registers in the MCUCR register are redirected to the 512-byte aligned start address.

Interrupt vector processing

The following code only LGT8F48D / 88D, for example, to explain the reset and interrupt vector programming, for reference only:

Assembly code example - LGT8F48D / 88D

address	Code	Description
0x000	RJMP RESET	Reset vector

0x001	RJMP EXT_INT0	External interrupt 0
0x002	RJMP EXT_INT1	External interrupt 1
0x003	RJMP PCINT0	Pin level change interrupt 0
0x004	RJMP PCINT1	Pin level change interrupt 1
0x005	RJMP PCINT2	Pin level change interrupt 2
0x006	RJMP WDT	The watchdog timer is interrupted
0x007	RJMP TIM2_COMPA	Timer 2 compare match group A interrupt
0x008	RJMP TIM2_COMPB	Timer 2 compare match group B interrupt
0x009	RJMP TIM2_OVF	Timer 2 overflow interrupt
0x00A	RJMP TIM1_CAPT	Timer 1 traps the interrupt
0x00B	RJMP TIM1_COMPA	Timer 1 compare match A group interrupt
0x00C	RJMP TIM1_COMPB	Timer 1 compare match group B interrupt
0x00D	RJMP TIM1_OVFR	Timer 1 overflow interrupt
0x00E	RJMP TIM0_COMPA	Timer 0 compare match group A interrupt
0x00F	RJMP TIM0_COMPB	Timer 0 compare match group B interrupt
0x010	RJMP TIM0_OVF	Timer 0 overflow interrupt
0x011	RJMP SPI_STC	SPI transfer complete interrupt
0x012	RJMP USART_RXC	USART reception complete interrupt
0x013	RJMP USART_UDRE	The USART data register is empty
0x014	RJMP USART_TXC	The USART sends an interrupt
0x015	RJMP ADC	ADC conversion complete interrupt
0x016	RJMP EE_RDY	The EEPROM controller is ready to interrupt
0x017	RJMP ANA_COMP	Comparator interrupt
0x018	RJMP TWI	TWI controller interrupt
0x019	NOP	Keep address
0x01A	NOP	Keep address
0x01B	RJMP PCI3	Pin level change interrupt 3
;		
0x01C (**the RESET:**)	LDI r16, high (RAMEND)	The main program starts
0x01D	OUT SPH, r16	Set the stack pointer to the top address of the RAM
0x01E	LDI r16, low (RAMEND)	
0x01F	OUT SPL, r16	
0x020	SEI	Enable global interrupts
0x021	...	

Page 49

Register definition

MCU Control Register **- MCUCR**

MCUCR - MCU control register

MCUCR: 0x35 (0x55) Default: 0x00

MCUCR	-	-	PUD	-	-	-	IVSEL	IVCE
R / W	-	-	R / W	-	-	-	R / W	R / W
Initial value	-	-	0	-	-	-	0	0

Bit definition

[0]	IVCE	Interrupt vector selection change enable bit, before changing IVSEL, you need to set this bit first After setting the IVSEL in 6 cycles.
[1]	IVSEL	Interrupt vector selection bit, this bit is set after the interrupt vector address will be based on the IVBASE register The value is mapped to the new address. For detailed mapping addresses, refer to the IVBASE register description
[2]	-	Keep not used
[3]	-	Keep not used

[4]	PUD	Global pull-up disabled bit
[5]	-	Keep not used
[6]	FPDEN	Flash Power / down enable control
[7]	FWKPEN	Fast wake-up mode enable control

Interrupt Vector Base Address Register - **IVBASE**

IVBASE - interrupt vector base address register

IVBASE: 0x75 Default: 0x00

IVBASE	IVBASE [7: 0]
R / W	R / W
Initial value	0x00

Bit definition

[7: 0]	IVBASE	If IVSEL is 1, the interrupt vector (except the reset vector) will be based on IVBASE at 512 Byte on the page to remap. The mapped interrupt vector base address is: (IVBASE << 8) + the corresponding vector address in Table 1

External interrupt

- 2 external interrupt sources
- Configurable level or edge trigger interrupt
- Can be used as a wake-up source in sleep mode

Overview

The external interrupt is triggered by the INT0 and INT1 pins. As long as the external interrupt is enabled, even if the two pins are configured for output Can also trigger an interrupt. This can be used to generate software interrupts. The external interrupt can be triggered by a rising edge, a falling edge, or a low level, It is configured by external interrupt control register EICRA. When an external interrupt is enabled and configured as a level trigger (only INT0 and INT1 pin), the interrupt will always be generated as long as the pin level is low. INT0 and INT1 pins are rising or falling The interrupt requires an IO clock to operate normally, while the INT0 and INT1 pin low-level interrupt are detected asynchronously. In addition to the idle mode, the IO clocks in other sleep modes are stopped. Therefore, these two external interrupts can be used as Wake-up source in other sleep modes except for Idle mode.

If the level-triggered interrupt is used as a wake-up source in power-saving mode, the changed level must be held for a certain amount of time to wake up MCU to reduce the MCU's sensitivity to noise. The required level must be kept long enough for the MCU to end the call Wake up the process, and then trigger the level interrupt.

Register definition

Register list

register	address	Defaults	description
			External interrupt control register A

EICRA	0x69	0x00	
EIMSK	0x3D	0x00	External interrupt mask register
EIFR	0x3C	0x00	External interrupt flag register

External Interrupt Control Register **A-EICRA**

EICRA - External Interrupt Control Register **A**

Address: 0x69 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	ISC11	ISC10	ISC01	ISC00
R / W	-	-	-	-	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 4	-	Keep it.
3	ISC11	INT1 pin interrupt trigger mode control bit high.
2	ISC10	INT1 pin interrupt trigger mode control bit low. When the global interrupt is set and the corresponding interrupt mask control bit of the GICR register is set, the external interrupt 1 Triggered by the INT1 pin. The trigger mode of the interrupt is described in the table. Before the edge detection MCU first mining

		The level on the INT1 pin. If you use the edge trigger mode or level change trigger mode, then A pulse with a duration greater than one system clock cycle will trigger an interrupt, and a short pulse can not guarantee Interrupted. If the low-level trigger mode is selected, then the low level must be maintained until the current instruction is completed Will trigger an interrupt.
1	ISC01	INT0 pin interrupt trigger mode control bit high.
0	ISC00	INT0 pin interrupt trigger mode control bit low. When the global interrupt is set and the corresponding interrupt mask control bit of the GICR register is set, the external interrupt 0 Triggered by the INT0 pin. The trigger mode of the interrupt is described in the table. Before the edge detection MCU first mining The level on the INT0 pin. If you use the edge trigger mode or level change trigger mode, then A pulse with a duration greater than one system clock cycle will trigger an interrupt, and a short pulse can not guarantee Interrupted. If the low-level trigger mode is selected, then the low level must be maintained until the current instruction is completed Will trigger an interrupt.

External interrupt 1 trigger mode see table below.

External interrupt 1 trigger mode control

ISC1 [1: 0]	description
0	External pin INT1 low trigger
1	Trigger on the rising or falling edge of external pin INT1
2	The falling edge of external pin INT1 is triggered
3	The rising edge of the external pin INT1 is triggered

External interrupt 0 trigger mode see table below.

External interrupt 0 Trigger mode control

ISC0 [1: 0]	description
0	External pin INT0 low level trigger
1	The rising edge of the external pin INT0 or the falling edge triggers
2	The falling edge of the external pin INT0 is triggered
3	The rising edge of the external pin INT0 is triggered

External Interrupt Mask Register **- EIMSK**

EIMSK - External interrupt mask register

Address: 0x3D Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	INT1	INT0
R / W	-	-	-	-	-	-	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 2	-	Keep it.
1	INT1	External pin 1 interrupt enable control bit. When the INT1 bit is set to "1" and the global interrupt is set, the external pin 1 interrupt is enabled and the wake

		Function is enabled. Even if the INT1 pin is configured as an output, as long as the pin level is changed accordingly The interrupt will be generated. When the INT1 bit is set to "0", the external pin 1 interrupt is disabled and the wake-up function is disabled.
0	INT0	External pin 0 interrupt enable control bit. When the INT0 bit is set to "1" and the global interrupt is set, the external pin 0 interrupt is enabled and the wake Function is enabled. Even if the INT0 pin is configured as an output, as long as the pin level has changed accordingly The interrupt will be generated. When the INT0 bit is set to "0", the external pin 0 interrupt is disabled and the wake-up function is disabled.

External Interrupt Flag Register - EIFR

EIFR - External interrupt flag register

Address: 0x3C Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	-	INTF1	INTF0
R / W	-	-	-	-	-	-	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 2	-	Keep it.
1	INTF1	External pin 1 interrupt flag bit. INTF1 is set when edge trigger external pin 1 is interrupted. When the low level triggers the external pin 1 When the interrupt is set, the INTF1 bit is not set. If the external pin 1 interrupt is enabled, the INT1EN bit is set to "1" When the global interrupt flag is set, an external pin 1 interrupt is generated. Execute this interrupt service routine when INTF1 This bit can also be cleared by clearing it automatically or by writing "1" to the INTF1 bit.
0	INTF0	External pin 0 interrupt flag. INTF0 is set when edge-triggered external pin 0 is interrupted. When low, the external pin 0 is triggered When the interrupt is set, the INTF0 bit is not set. If the external pin 0 interrupt is enabled, the INT0EN bit is set to "1" When the global interrupt flag is set, an external pin 0 interrupt is generated. Execute this interrupt service routine when INTF0 This bit can also be cleared by clearing it automatically or by writing "1" to the INTF0 bit.

Input / output subsystem

Overview

All MCUs based on the LGT8 core family have I / O port read-change-write functions. This means that one by one
The status of the port can be changed individually using the SBI and CBI instructions without affecting any other I / O. Again, change
The direction of a port or control its pull-up resistor can also be the case.

Most of the I / O of the LGT8FX8D has symmetrical drive characteristics that can drive and absorb large currents. I / O has two
Level drive capability, the user can control the drive capability of each group of I / O. I / O drive capability can directly drive some LEDs.

LGT8FX8D most of the I / O can drive up to 30mA of current, can be directly used to drive segment code LED.

All I / O VCC and GND directly have independent ESD protection diodes, designed to withstand at least up to
5000V ESD pulse.

LGT8FX8D series most of the ***I / O*** internal has a default uncontrollable weak pull ***(*** about ***80KΩ)*** , when the ***I / O*** work
For input ***I / O*** , this default weak pull-up is forced on. The weak pull-up of the input mode ***I / O*** favors the low power mode
Control, external floating input mode ***I / O*** , if no pull-up / pull-down processing, will bring additional leakage.

When the ***I / O*** work in the output mode, or analog function, the default weak pull-up automatically shut down.

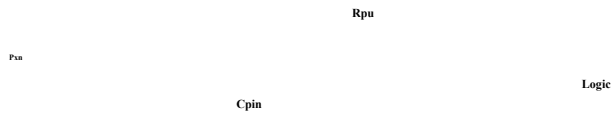
Not all ***I / Os*** has a default weak pull-down, do not have the default weak pull-up ***I / O*** include:

PD2 , ***PD3*** , ***PD5*** , ***PB1*** , ***PB2*** , ***PE0*** , ***PE2***

If these ***I / O*** work in the input ***I / O*** mode, before entering the low power mode, the software needs to write through the corresponding ***PORT***
The register is ***1*** to open the internal controllable strong pull-up.

PE0 / 2 default to ***SWD*** interface needs to be disabled ***PE0 / 2*** of the ***SWD*** function after, before opening the internal strong pullup.

I / O equivalent circuit diagram:



All of the following registers in this chapter are described in a uniform way. The lowercase "x" indicates the letter number of the port, the lower case "n"
Indicates the bit number in the port. However, when using the port register in the program, you must use the exact register name. such as
PORTB3, which represents the third bit of PORTB, where it is unified with PORTxn said. A detailed definition of I / O related registers,
Please refer to the Register Description section.

Each port is assigned three I / O register spaces, which are: Port Data Output Register (PORTx), Port Direction
Register (DDRx), port data input register (PINx). The port data input register is a read-only register. Data output
The register and port direction registers are readable or rewritten. The PUD bit in the MCUCR register is used to control all I / O
Pull-up resistor, when the PUD bit is 1, will disable the I / O pull-up resistor.

Page 54

Most I / O in addition to the general input / output function, will be reused for other peripheral functions. Specific reusable work
Please refer to the chapter on port function reuse.

It should be noted that enabling multiple port reuse does not affect these ports as digital I / O. and
Some of the alternate functions may also need to control the input / output direction of the port via the I / O register. The specific settings will be in each
A description of the documentation for the multiplex module.

Universal input / output port

As a general purpose I / O, the port is a bidirectional drive I / O port with an internal programmable pull-up.

The following figure shows the equivalent circuit diagram of the general purpose I / O port:

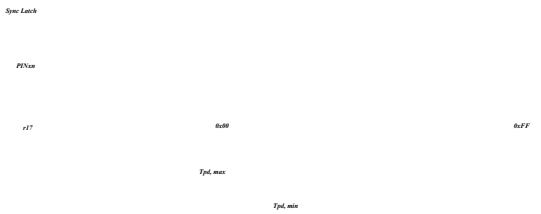
PUD: PULLUP DISABLE	WDx: WRITE DDRx
SLEEP: SLEEP CONTROL	RDx: READ DDRx
IO_CLK: I / O CLOCK	WRx: WRITE PORTx
	RRx: READ PORTx REGISTER
	RPx: READ PORTx PIN
	WPx: WRITE PINx REGISTER

Port usage configuration

Each port is controlled by three register bits: DDxn, PORTxn and PINxn. Where DDxn is available for use via DDRx
Register access, PORTxn can be accessed via the PORTx register, and PINxn can be accessed via the PINx register.

The DDRxn register bits are used to set the input / output direction of the port. If DDxn is set to 1, the Pxn port is assigned
Set to an output port. If DDxn is set to 0, Pxn is configured as an input port.

Page 55



If the PORTxn bit is written 1 and the port is configured as an input port, the pull-up resistor for this port is asserted.
If you want to disable the pull-up resistor on the port, PORTxn must be either 0 or configured as an output port.

The reset status of the port is the input state and the pull-up resistor is invalid.

PORTxn is set to 1, and the port is configured as an output port and the external port will be driven high. Such as
If PORTxn is set to 0, the port will be driven low.

Input / output switch

When the I / O state is between the tri-state ([DDxn, PORTxn]) = 0b00) and the output high ([DDxn, PORTxn] = 0b11)
In case of time, there will be a port pull-up or output to a low intermediate state. In general, the pull-up resistor can be accepted,
Because in a high resistance environment, the drive between the high and the difference between the pull is not important. If this is not the case, you can pass
The PUD bit in the MCUCR register is turned off so the pull-up function of the port.

Likewise, the same problem arises when switching between the pull-up enable input and the output low. The user must make
([DDxn, PORTxn] = 0b00) or the output high ([DDxn, PORTxn] = 0b11) as the intermediate state.

Port driver configuration table:

DDxn	PORTxn	PUD	Port state	pull up	Function Description
0	0	X	enter	Forbidden	Tri-state
0	1	0	enter	Enable	If the external pull-down, the pin will fan out the current
0	1	1	enter	Forbidden	Tri-state
1	0	X	Output	Forbidden	Output low (fan-in)
1	1	X	Output	Forbidden	Output high (fanout)

Read port value

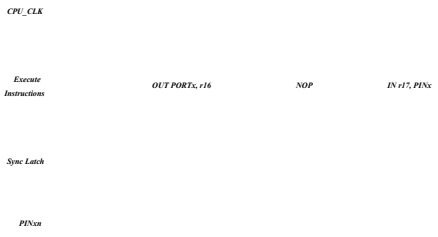
Regardless of how the port direction bit DDxn is set, the current status of the port can be read via the PINxn register bit.
To avoid directly reading the metastable state generated by the port, the PINxn register bit is the result of the port passing through a synchronizer. Synchronizer
Is composed of a latch and a register, so there is a small delay between the value of PINxn and the current port.
This delay is due to the presence of the synchronizer, with a delay of up to one half of the system cycle.

The delay time is Tpd, max and Tpd, min in the following figure:

We assume that the system cycle from the system clock on the first falling edge of the latch when the clock is low when the data,
When the clock is high, the data passes through the latch, as shown in the shaded area above. When the clock is low, the port data is locked
And the rising edge of the next clock is registered into the PINxn register. In the figure above, Tpd, max and Tpd, min

The maximum and minimum delays for port data are divided into 1.5 cycles and 0.5 cycles.

If you want to read the port value set by the software, you need to insert an empty operation instruction in the I / O write and read bytes support
(NOP). The timing is shown below:



r17 *0x00* *0xFF*

Tpd

The following code shows how to set port 0 to pin 0/1 to be high, 2/3 is low, define pins 4 to 7 for input and
The pull-up resistors of pins 6, 7 are enabled. And then the value of the pin back to read the general working register, according to the previous description,
A NOP instruction is inserted directly on the pin's output and input.

Assembly code

```
; Define Pull-ups and set up high
; Define directions for port pins
LDI r16, (1 << PB7) | (1 << PB6) | (1 << PB1) | 1 << PB0)
LDI r17, (1 << DDB3) | (1 << DDB2) | (1 << DDB1) | (1 << DDB0)
OUT PORTB, r16
OUT DDRB, r17
; Insert nop for synchronization
NOP
; Read port pins
IN r16, PINB
```

C language code

```
unsigned char
/ * Define pull-ups and set up high * /
/ * Define directions for port pins * /
PORTB = (1 << PB7) | (1 << PB6) | (1 << PB1) | (1 << PB0);
DDRB = (1 << DDB3) | (1 << DDB2) | (1 << DDB1) | (1 << DDB0);
/ * Insert nop for synchronization * /
__no_operation ();
/ * Read port pins * /
I = PINB;
```


Input enable and sleep control

From the I / O equivalent circuit diagram we can see that the digital input can be clamped to the ground under the control of the SLEEP signal
Level. The SLEEP signal is controlled by the MCU's sleep controller and various sleep modes. This ensures that you are going to sleep
, The system will not be due to the port input floating caused by leakage.

The SLEEP control of the port is replaced by an external interrupt function. If the external interrupt request is invalid, SLEEP control remains
It can work. SLEEP control function will be replaced by some other second function, please refer to the following on the port
Two functions introduced.

Idle port processing

If some ports are not being used, it is recommended to drive them to a fixed level. In any case, floating
The pin will bring more power consumption, and will lead to the system under strong interference becomes unstable.

The easiest way to give a fixed level to a port is to turn on the pull-up resistor on the port. It should be noted that the pull-up resistor
It is forbidden during power-on reset. Pull-up resistor will also bring the excess leakage. It is therefore recommended to use an external one
Pull or pull-down resistor connection. It is not advisable to connect the port directly to the power supply or ground because if these pins are configured as
Output, it may lead to a very large current through the port, the chip caused a devastating impact.

Port reuse function

Most of the ports have a multiplexing function, the following equivalent circuit describes the port multiplexing function of the port control. These complex
The function does not necessarily exist with the port pin.

PUOExn: Pxn PULL-UP OVERRIDE ENABLE	PUD: PULLUP DISABLE
PUOVxn: Pxn PULL-UP OVERRIDE VALUE	WDx: WRITE DDRx

DDOExn: Pxn DATA DIRECTION OVERRRIDE ENABLE	RDx: READ DDRx
DDOVxn: Pxn DATA DIRECTION OVERRIDE VALUE	RRx: READ PORTx REGISTER
PVOExn: Pxn PORT VALUE OVERRIDE ENABLE	WRx: WRITE PORTx
PVOVxn: Pxn PORT VALUE OVERRIDE VALUE	RPx: READ PORTx PIN
DIEOExn: Pxn INPUT-ENABLE OVERRIDE ENABLE	WPx: WRITE PINx
DIEOVxn: Pxn INPUT-ENABLE OVERRIDE VALUE	IO_CLK: I / O CLOCK
SLEEP: SLEEP CONTROL	DIxn: INPUT PIN n ON PORTx
PTOExn: Pxn PORT TOGGLE OVERRIDE ENABLE	AIOxn: ANALOG I / O PIN n ON PORTx

Alternate function control signal General description:

Signal full name	Functional description
PUOE pull-up multiplex enable	This bit is 1, pull-up enable is controlled by PVOV; if this bit is 0, pull-up enable by DDxn, PORTxn and PUD common control system
PUOV pull-up multiplexing value	If PUOE is 1, this bit will enable the pull-up of the pin Resistance, otherwise it will disable the pin pull-up resistor
DDOE port direction multiplexing enable	Bit is 1, pin output enabled by DDOE control, otherwise by DDxn control
DDOV port direction multiplexing value	If DDOE is 1 and bit 1 is 1, the output of the pin is enabled Function, otherwise the output of the pin is turned off
PVOE port data multiplexing enable	If the bit is 1, and the pin output is enabled, the pin is output The value will be controlled by PVOV, otherwise it will be controlled by PORTxn
PVOV port data reuse value	Refer to the PVOE function description
PTOE port rollover is enabled	The bit is 1, and the PORTxn bit will flip
DIEOE digital input enable multiplexing enable	If the bit is 1, the port digital input enable will be controlled by DIEOV System; otherwise there will be MCU running state control
DIEOV digital input enables multiplexed values	If DIEOE is 1, the port's digital input function will be times Bit control, independent of the MCU operating status

DI	Digital input	This is the digital input signal input to the alternate function module. From the I / O and other circuit can be seen in the next, this value in the dense After the special trigger, but before the I / O input synchronizer. This one Signal is connected to the peripheral module, the peripheral module will be required To be synchronized
AIO	Analog input	Analog input / output signal, this signal directly with I / O PAD Connected, can be used as a simulation of the two-way signal. This signal is straight Connected to the internal ADC, comparator and other analog modules port phase connection

The following section will briefly describe the alternate function of each pin and the associated control signal.

Port **B** reuse function

Pin multiplexing function description

PB7	XTAL2 / TOSC2 (external main crystal pin 2) PCINT7 (pin change interrupt 7)
PB6	XTAL1 / TOSC1 (external main crystal pin 1) PCINT6 (pin change interrupt 6)
PB5	SCK (SPI bus master clock input) PCINT5 (pin change interrupt 5)
PB4	MISO (SPI bus master input / slave output) PCINT4 (pin change interrupt 4)
PB3	MOSI (SPI bus master output / slave input) OC2A (Timer / Event Counter 2 Compare Match Output A) PCINT3 (pin change interrupt 3) SSN (SPI bus slave device select input)
PB2	OC1B (Timer / Event Counter 1 Compare Match Output B) PCINT2 (pin change interrupt 2)
PB1	OC1A (Timer / Event Counter 1 Compare Match Output A) PCINT1 (pin change interrupt 1) ICP1 (Timer / Event Counter 1 capture input)
PB0	CLKO (system clock output) PCINT0 (pin change interrupt 0)

XTAL2 / TOSC2 / PCINT7 - Port *B* pin *7*

XTAL2 : External crystal pin 2. When used as a crystal clock signal, this pin will not be used as an I / O.

TOSC2 : Timer external crystal pin 2. When the internal RC is configured as the main working clock of the chip, and the difference is enabled Step timer function (ASSR register configuration), this pin will be used as a timer external crystal pin. When ASSR is registered AS2 is set to 1 and EXCLK is set to 0, the Timer / Event Counter 2 is enabled with an external crystal Clock function, PB7 will be disconnected from the internal I / O port and become the reverse output pin of the internal oscillator amplifier. This mold The external crystal is connected to the pin.

PCINT7 : Pin Change Interrupt 7. PB7 is an external interrupt source.
If PB7 is used for crystal pins, the values of DDB7, PORTB7 and PINB7 will not make any sense.

XTAL1 / TOSC1 / PCINT6 - Port *B* pin *6*

XTAL1 : External crystal pin 1.

TOSC1 : Timer external crystal pin 1. When the internal RC is configured as the main working clock of the chip, and the difference is enabled Step timer function (ASSR register configuration), this pin will be used as a timer external crystal pin. When ASSR is registered

AS2 is set to 1 and EXCLK is set to 0, the Timer / Event Counter 2 is enabled with an external crystal

Clock function, PB6 will be connected to the internal I / O port port as the internal oscillator amplifier input pin. In this mode,

The external crystal is connected to the pin.

PCINT6 : Pin Change Interrupt 6. PB6 is an external interrupt source.

If PB6 is used for the crystal pin, the values of DDB6, PORTB6 and PINB6 will have no meaning.

SCK / PCINT5 - Port *B* pin *5*

SCK : SPI controller master device clock output, slave device clock input. When the SPI controller is configured as a slave device,

This pin will be configured as an input pin, not controlled by DDB5. When the SPI controller is configured as a master device,

The direction of this pin is controlled by DDB5. When this pin is forced by the SPI input, it can still pass PORTB5

Bit control pull-up resistor.

PCINT5 : Pin level change interrupt. PB5 is an external interrupt source.

MISO / PCINT4- Port *B* pin *4*

MISO : SPI control master device data input, slave device data output. When the SPI is configured as a master device, this pin will be

Will be forced to input, not subject to DDB4 control. When the SPI is used as a slave device, this pin is the data side

To control by DDB4. When this pin is forced by the SPI controller, its pull-up resistor can still pass

PROTB4 control.

PCINT4 : Pin level change interrupt. PB4 is an external interrupt source.

MOSI / OC2A / PCINT3- Port *B* pin *3*

MOSI : SPI controller master device data output, slave device data input. When the SPI is configured as a slave, this pin

Will be forced to input, and not subject to DDB3 control. When the SPI controller is configured as a master device, this pin is

The method is controlled by DDB3. When this pin is forced by the SPI control input, it can still be controlled by PORTB3

Of the pull-up resistor.

OC2A : Timer / Counter 2 Group A compare match output. PB3 can be used as a timer / counter 2 compare match

Department of output. The pin must be set to output via DDB3. At the same time, OC2A is also the PWM mode of Timer 2

Type output pin.

PCINT3 : Pin level change interrupt. PB3 is an external interrupt source.

SSN / OC1B / PCINT2 - Port *B* pin *2*

SSN : SPI slave device chip select input. When the SPI controller is configured as a slave, this pin will be forced for input,

And is not controlled by DDB2. As a slave device, the SPI controller is driven low on the SSN is valid. When the SPI controls

The device is configured as the master device and the direction of this pin is controlled by DDB2. When this pin is forced by the SPI controller to lose

After entering, you can still control the pull-up resistor via PORTB2.

OC1B : B / O of the Timer / Event Counter 1 compare match output. PB2 can be used as a timer / counter 1 compare match

Department of output. The pin must be set to output via DDB2. At the same time, OC1B is also the PWM mode of Timer 1

Type output pin.

PCINT2 : Pin level change interrupt. PB2 is an external interrupt source.

OC1A / PCINT1 - Port *B* pin *1*

OC1A : Group A compare match output for Timer / Event Counter 1. PB1 can be used as a timer / counter 1 compare match

Department of output. The pin must be set to output via DDB1. At the same time, OC1A is also the PWM mode of Timer 1

Type output pin.

PCINT1 : Pin level change interrupt. PB1 is an external interrupt source.

Page 61

ICP1 / CLKO / PCINT0 - Port *B* pin *0*

ICP1 : Capture input pin for Timer / Event Counter 1

CLKO : System operating clock output, when CLKPR register CLKOE bit is 1, this pin will be forced to Output, not controlled by DDB0. The output frequency is the operating clock frequency of the current system.

PCINT0 : Pin level change interrupt. PB0 is an external interrupt source.

PB7 ... PB4 multiplexing control logic table

Signal name	PB7 / XTAL2 / TOSC2 / PCINT7	PB6 / XTAL1 / TOSC1 / PCINT6	PB5 / SCK PCINT5	PB4 / MISO PCINT4
PUOE	OSCEN | AS2	OSCEN | AS2	SPE & MSTR	SPE & MSTR
PUOV	0	0	PORTB5 & PUD	PORTB4 & PUD
DDOE	OSCEN | AS2	OSCEN | AS2	SPE & MSTR	SPE & MSTR
DDOV	0	0	0	0
PVOE	0	0	SPE & MSTR	SPE & MSTR
PVOV	0	0	SCK Output	SPI Slave Output
DIEOE	PCINT7 Enable	PCINT6 Enable	PCINT5 Enable	PCINT4 Enable
DIEOV	1	1	1	1
DI	PCINT7 Input	PCINT6 Input	PCINT5 Input SCK Input	PCINT4 Input SPI Master Input
AIO	XTAL2 TOSC2	XTAL1 TOSC1	-	-

[Explanation]: OSCEN includes OSCK_EN and OSCM_EN, please refer to PMCR register description

PB3 ... PB0 Alternate function control logic table

Signal name	PB3 / MOSI / OC2A / PCINT3	PB2 / SSN / OC1B / PCINT2	PB1 / OC1A / PCINT1	PB0 / ICP1 / CLKO / PCINT0
PUOE	SPE & MSTR	SPE & MSTR	0	0
PUOV	PORTB3 & PUD	PORTB2 & PUD	0	0
DDOE	SPE & MSTR	SPE & MSTR	0	CLKO ENABLE
DDOV	0	0	0	1
PVOE	SPE & MSTR + OC2A ENABLE	OC1B ENABLE	OC1A ENABLE	CLKO ENABLE
PVOV	SPI Master Output OC2A	OC1B	OC1A	CLKO
DIEOE	PCINT3 Enable	PCINT2 Enable	PCINT1 Enable	PCINT0 Enable
DIEOV	1	1	1	1
DI	PCINT3 Input SPI Slave Input	PCINT2 Input SPI Slave Select	PCINT1 Input	PCINT0 Input ICP1 Input
AIO	-	-	-	-

Page 62

Port **C** multiplexing function

Pin	Alternate function description
PC6	RESETN (external reset input)
	PCINT14 (pin change interrupt 14)
PC5	ADC5 (ADC input channel 5)
	SCL (TWI clock line)
	PCINT13 (pin change interrupt 13)
PC4	ADC4 (ADC input channel 4)
	SDA (TWI data cable)
	PCINT12 (pin change interrupt 12)
PC3	ADC3 (ADC input channel 3)
	PCINT11 (pin change interrupt 11)
PC2	ADC2 (ADC input channel 2)
	PCINT10 (pin change interrupt 10)
PC1	ADC1 (ADC input channel 1)
	PCINT9 (pin change interrupt 9)
PC0	ADC0 (ADC input channel 0)
	PCINT8 (pin change interrupt 8)

RESETN / PCINT4- Port *C* pin *6*

RESETN : External reset input pin. After a power-on reset, this pin defaults to an external reset function. Can pass through IOCR
The register turns off the external reset function. When the external reset function is turned off, this pin can be used as a general purpose I / O. But need to
It should be noted that in the power and other reset process, this pin defaults to reset input, so if the user needs
Use this pin of the general I / O function, the external circuit can not affect the chip power and reset process, it is recommended that this
The pins are configured as I / O for the output function and an external pull-up resistor is added externally.

PCINT14 : Pin level change interrupt. After turning off the external reset input function of this pin, PC6 can be done outside
Interrupt source.

SCL / ADC5 / PCINT13 - Port *C* pin *5*

SCL : TWI interface clock signal. After the TWEN bit in the TWCR register is set, the TWI interface is enabled and PC5 will be enabled
TWI control, become TWI interface clock signal.

ADC5 : ADC input channel 5. The DIDR register is used to turn off digital functions of the M / A I / O to avoid digital parts
On the impact of analog circuits. Please refer to the ADC-related section.

PCINT13 : Pin Change Interrupt 13.

SDA / ADC4 / PCINT12 - Port *C* pin *4*

SDA : TWI interface data signal. After the TWEN bit in the TWCR register is set, the TWI interface is enabled and PC4 will be enabled
TWI control, as TWI interface data signal.

ADC4 : ADC input channel 4. The DIDR register is used to turn off digital functions of the M / A I / O to avoid digital parts
On the impact of analog circuits. Please refer to the ADC-related section.

PCINT12 : Pin Change Interrupt 12.

ADC3 / PCINT11 - Port *C* pin *3*

ADC3 : ADC input channel 3. The DIDR register is used to turn off digital functions of the M / A I / O to avoid digital parts
On the impact of analog circuits. Please refer to the ADC-related section.

PCINT11 : Pin change interrupt 11.

ADC2 / PCINT1 - Port *C* pin *2*

ADC2 : ADC input channel 2. The DIDR register is used to turn off digital functions of the M / A I / O to avoid digital parts

On the impact of analog circuits. Please refer to the ADC-related section.
PCINT10 : Pin change interrupt 10.

ADC1 / PCINT9 - Port *C* pin *1*

ADC1 : ADC input channel 1. The DIDR register is used to turn off digital functions of the M / A I / O to avoid digital parts
On the impact of analog circuits. Please refer to the ADC-related section.

PCINT9 : Pin Change Interrupt 9.

ADC0 / PCINT8 - Port *C* pin *0*

ADC0 : ADC input channel 0. The DIDR register is used to turn off digital functions of the M / A I / O to avoid digital parts
On the impact of analog circuits. Please refer to the ADC-related section.

PCINT8 : Pin change interrupt 8.

PC6 ... PC4 multiplexing control logic table

Signal name	PC6 / RESETN / PCINT14	PC5 / ADC5 / TK9 / SCL / PCINT13	PC4 / ADC4 / TK8 / SDA / PCINT12
PUOE	RSTIOEN	TWI Enable	TWI Enable
PUOV	1	PORTC4 & PUD	PORTC4 & PUD
DDOE	RSTIOEN	TWI Enable	TWI Enable
DDOV	0	SCL Output	SDA Output
PVOE	RSTIOEN	TWI Enable	TWI Enable
PVOV	1	0	0
DIEOE	PCINT14 Enable + RSTIOEN	PCINT13 Enable + TWI Enable	PCINT12 Enable + TWI Enable
DIEOV	1	1	1
DI	PCINT14 Input External Reset Input	PCINT13 Input SCL Input	PCINT12 Input SDA Input
AIO	-	ADC5	ADC4

PC3 ... PC0 multiplexing control logic table

Signal name	PC3 / ADC3 / PCINT11	PC2 / ADC2 / PCINT10	PC1 / ADC1 / PCINT9	PC0 / ADC0 / PCINT8
PUOE	0	0	0	0
PUOV	0	0	0	0

DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
DIEOE	PCINT11 Enable	PCINT10 Enable	PCINT9 Enable	PCINT8 Enable
DIEOV	1	1	1	1
DI	PCINT11 Input	PCINT10 Input	PCINT9 Input	PCINT8 Input
AIO	ADC3	ADC2	ADC1	ADC0

Port **D** multiplexing function

Pin	Alternate function description
PD7	AIN1 (analog comparator negative input) PCINT23 (pin change interrupt 23)
PD6	AIN0 (analog comparator positive input) OC0A (Timer / Event Counter 0 Compare Match Output A) PCINT22 (pin change interrupt 22)

	T1 (Timer / Counter 1 external count clock input)
PD5	OC0B (Timer / Event Counter 0 Compare Match Output B)
	PCINT21 (pin change interrupt 21)
	XCK (USART external clock input / output)
PD4	T0 (Timer / Event Counter 0 External Count Clock Input)
	PCINT20 (pin change interrupt 20)
	INT1 (external interrupt input 1)
PD3	OC2B (Timer / Event Counter 2 Compare Match Output B)
	PCINT19 (pin change interrupt 19)
PD2	INT0 (external interrupt input 0)
	PCINT18 (pin change interrupt 18)
PD1	TXD (USART data output)
	PCINT17 (pin change interrupt 17)
PD0	RXD (USART data entry)
	PCINT16 (pin change interrupt 16)

AIN1 / OC2B / PCINT23 - Port *D* pin *7*

AN1 : Analog Comparator Negative Input. The digital input function of the PD7 pin is turned off by the DIDR1 register and is turned off Port pull-up resistor to avoid digital port interference to the analog circuit.

OC2B : B / O of the Timer / Event Counter 2 compare match output. PD7 can be used as a timer / counter 2 compare match Department of output. The pin must be set to output via DDD7. At the same time, OC2B is also the PWM mode of Timer 2 Type output pin.

PCINT23 : Pin Change Interrupt 23.

AIN0 / OC0A / PCINT22- Port *D* pin *6*

AN0 : Analog Comparator Positive Input. The digital input function of the PD6 pin is turned off by the DIDR1 register and is turned off

Port pull-up resistor to avoid digital port interference to the analog circuit.

OC0A : Timer / Counter 0 Group A compare match output. PD6 can be used as a timer / counter 0 compare match Department of output. The pin must be set to output via DDD6. At the same time, OC0A is also the PWM mode of Timer 0 Type output pin.

PCINT22 : Pin change interrupt 22.

T1 / OC0B / PCINT21 - Port *D* pin *5*

T1 : External count clock input for Timer / Event Counter 1

OC0B : Timer / Counter 0 Group B compare match output. PD5 can be used as a timer / counter 0 compare match Department of output. In this case, the pin must be set to output via DDD5. At the same time, OC0B is also the PWM mode of Timer 0 Type output pin.

PCINT21 : Pin change interrupt 21.

XCK / T0 / PCINT20 - Port *D* pin *4*

XCK : Synchronous mode USART external clock signal

T0 : External count clock input for Timer / Event Counter 0

PCINT20 : Pin change interrupt 20.

INT1 / OC2B / PCINT19 - Port *D* pin *3*

INT1 : External interrupt input 1

OC2B : B / O of the Timer / Event Counter 2 compare match output. PD3 can be used as a timer / counter 2 compare match Department of output. In this case, the pin must be set to output via DDD3. At the same time, OC2B is also the PWM mode of Timer 2 Type output pin.

PCINT19 : Pin Change Interrupt 19.

INT0 / PCINT18 - Port *D* pin *2*

INT0 : External interrupt input 0

PCINT18 : Pin change interrupt 18.

TXD / PCINT17 - Port *D* pin *1*

TXD : Transfer data (USART data output). After the USART transmitter is enabled, PD1 will be forced to output DDD1 control.

PCINT17 : Pin Change Interrupt 17.

RXD / PCINT16 - Port *D* pin *0*

RXD : Transfer data (USART data entry). After the USART receiver is enabled, PD0 will be forced as input DDD0 control. When the pin is forced into the input by USART, the pull-up resistor can still be controlled by the PORTD0 bit.

PCINT16 : Pin change interrupt 16.

PD7 ... PD4 multiplexing control logic table:

Signal name	PD7 / AIN1 / PCINT23	PD6 / AIN0 / OC0A / PCINT22	PD5 / OC0B / PCINT21	PD4 / XCK / T0 / PCINT20
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	OC0AEN & OC0AS	OC0B Enable	XCKOEN
PVOV	0	OC0A	OC0B	XCK Output
DIEOE	PCINT23 Enable	PCINT22 Enable	PCINT21 Enable + T1EN	PCINT20 Enable + XCKIEN + T0EN
DIEOV	1	1	1	1
DI	PCINT23 Input	PCINT22 Input	PCINT21 Input T1 Input	PCINT20 Input XCK Input T0 Input
AIO	-	-	-	-

PD3 ... PD0 multiplexing control logic table:

Signal name	PD3 / OC2B / INT1 / PCINT19	PD2 / INT0 / PCINT18	PD1 / TXD / PCINT17	PD0 / RXD / PCINT16
PUOE	0	0	TXEN	RXEN
PUOV	0	0	0	PORTD0 & PUD
DDOE	0	0	TXEN	RXEN
DDOV	0	0	1	0
PVOE	OC2B Enable	0	TXEN	0
PVOV	OC2B	0	TXD	0
DIEOE	PCINT19 Enable + INT1 Enable	PCINT18 Enable + INT0 Enable	PCINT17 Enable	PCINT16 Enable + RXEN
DIEOV	1	1	1	1
DI	PCINT19 Input INT1 Input	PCINT18 Input INT0 Input	PCINT17 Input	PCINT16 Input RXD
AIO	-	-	-	-

Port **E** multiplexing function

Pin	Alternate function description
PE6	VREF (ADC external reference voltage) PCINT30 (pin change interrupt 30)
PE5	CLKO (system clock output) PCINT29 (pin change interrupt 29)
PE4	OC0A (Timer / Event Counter 0 Compare Configuration Output A) PCINT28 (pin change interrupt 28)
PE3	ADC7 (ADC input channel 7)

Page 67

PE2	PCINT27 (pin change interrupt 27) SWD (SWD debugger data cable) PCINT26 (pin change interrupt 26)
PE1	ADC6 (ADC input channel 6) PCINT25 (pin change interrupt 25)
PE0	SWC (SWD debugger clock input) PCINT24 (pin change interrupt 24)

VREF / PCINT30 - Port *E* pin *6*

VREF : ADC external reference power input, used as an analog function, the need to set the corresponding digital I / O input, And turn off the pull-up resistor to prevent digital circuits from interfering with analog circuits.

PCINT30 : Pin change interrupt 30

CLKO / PCINT29 - Port *E* pin *5*

CLKO : This function is the same as the CLKO function of PB0. Can be used as a backup pin for PB0 / CLKO

PCINT29 : Pin Change Interrupt 29

OC0A / PCINT28 - Port *E* pin *4*

OC0A : Timer / Counter 0 Group A compare match output. PE4 can be used as a timer / counter 0 compare match Department of output. The pin must be set to output via DDE4. At the same time, OC0A is also the PWM mode of Timer 0 Type output pin.

PCINT28 : Pin change interrupt 28

ADC7 / PCINT27 - Port *E* pin *3*

ADC7 : ADC input channel 7. The DIDR register is used to turn off digital functions of the M / A I / O to avoid digital parts On the impact of analog circuits. Please refer to the ADC-related section.

PCINT27 : Pin change interrupt 27

SWD / PCINT26 - Port *E* pin *2*

SWD : SWD debugger data cable. After the system power-on reset, PE2 defaults to SWD function. The user can pass by MCUSR register SWDD bit is set to turn off the SWD debugger function. SWD is turned off, the debugging function will not be possible use.

PCINT26 : Pin change interrupt 26

ADC6 / PCINT25 - Port *E* pin *1*

ADC6 : ADC input channel 6. The DIDR register is used to turn off digital functions of the M / A I / O to avoid digital parts On the impact of analog circuits. Please refer to the ADC-related section.

PCINT25 : Pin change interrupt 25

SWC / PCINT24 - Port *E* pin *0*

SWC : SWD debugger clock line. After the system power-on reset, PE0 defaults to SWC function. The user can pass by MCUSR register SWDD bit is set to turn off the SWD debugger function. SWD is turned off, the debugging function will not be possible use.

PCINT24 : Pin change interrupt 24

PE6 ... PE4 multiplexing control logic table

Signal name	PE6 / VREF / PCINT30	PE5 / CLKO / PCINT29	PE4 / OC0A / PCINT28
PUOE	REFIOEN	0	0
PUOV	0	0	0
DDOE	REFIOEN	CLKO Enable 1	0
DDOV	0	1	0
PVOE	REFIOEN	CLKO Enable 1	OC0AEN & OC0AS
PVOV	1	CLKO	OC0A
DIEOE	PCINT30 Enable + REFIOEN	PCINT29 Enable	PCINT28 Enable
DIEOV	1	1	1
DI	PCINT30 Input	PCINT29 Input	PCINT28 Input
AIO	VREF	-	-

PE3 ... PE0 Multiplexed control logic table:

Signal name	PE3 / ADC7 / TK11 / PCINT27	PE2 / TK7 / SWD / PCINT26	PE1 / ADC6 / TK10 / PCINT25	PE0 / TK6 / SWC / PCINT24
PUOE	0	SWDD	0	SWDD
PUOV	0	1	0	1
DDOE	0	SWDD	0	SWDD
DDOV	0	SWD Output	0	0
PVOE	0	SWDD	0	0
PVOV	0	0	0	0
DIEOE	PCINT27 Enable	PCINT26 Enable + SWDD	PCINT25 Enable	PCINT24 Enable + SWDD
DIEOV	1	1	1	1
DI	PCINT27 Input	PCINT24 Input SWD Input	PCINT25 Input	PCINT24 Input SWC Input
AIO	ADC7	-	ADC6	-

Page 69

Register definition

MCU Control Register - **MCUCR**

MCUCR - **MCU** control register

MCUCR: 0x35 (0x55) Default: 0x00

MCUCR	FWKPEN	FPDEN		PUD	-	-	-	IVSEL	IVCE
R / W	R / W	R / W		R / W	-	-	-	R / W	R / W
Initial value	0	0		0	-	-	-	0	0

Bit definition

[0]	IVCE	Interrupt Vector Select the Change Enable bit
[1]	IVSEL	Interrupt vector selection bit
[2]	-	Keep not used
[3]	-	Keep not used
[4]	PUD	Global pull-up disabled bit
[5]	-	Keep not used
[6]	FPDEN	FLASH power down control
[7]	FWKPEN	Fast wake mode control

Port **B** Output Data Register - **PORTB**

PORTB - Port **B** Output Data Register

PORTB: 0x05 (0x25) Default: 0x00

PORTB	PORTB7	PORTB6	PORTB4	PORTB1	PORTB1	PORTB1	PORTB1	PORTB1
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	0	0	0	0	0	0	0	0

[0]	PORTB0	Port B outputs bit 0
[1]	PORTB1	Port B outputs bit 1
[2]	PORTB2	Port B outputs bit 2
[3]	PORTB3	Port B outputs bit 3
[4]	PORTB4	Port B outputs bit 4
[5]	PORTB5	Port B outputs bit 5
[6]	PORTB6	Port B outputs bit 6
[7]	PORTB7	Port B outputs bit 7

Port **B** Direction Register - **DDRB**

DDRB - port **B** direction register

DDRB: 0x04 (0x24) Default: 0x00

Page 70

DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	0	0	0	0	0	0	0	0

[0]	DDB0	PB0 direction control bit

[1]	DDB1	PB1 direction control bit
[2]	DDB2	PB2 direction control bit
[3]	DDB3	PB3 direction control bit
[4]	DDB4	PB4 direction control bit
[5]	DDB5	PB5 direction control bit
[6]	DDB6	PB6 direction control bit
[7]	DDB7	PB7 direction control bit

Port **B** Input Data Register **- PINB**

PINB - Port **B** input data register

PINB: 0x03 (0x23) Default: 0x00

PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	0	0	0	0	0	0	0	0

[0]	PINB0	PB0 port data
[1]	PINB1	PB1 port data
[2]	PINB2	PB2 port data
[3]	PINB3	PB3 port data
[4]	PINB4	PB4 port data
[5]	PINB5	PB5 port data
[6]	PINB6	PB6 port data
[7]	PINB7	PB7 port data

Port **C** Output Data Register **- PORTC**

PORTC - Port **C** output data register

PORTC: 0x08 (0x28) Default: 0x00

PORTC	-	PORTC6	PORTC5	PORTC4	PORTC3	PORTC1	PORTC1	PORTC1	PORTC
R / W	-	R / W	R / W	R / W	R / W	R / W	R / W	R / W	
Initial value	-	0	0	0	0	0	0	0	

[0]	PORTC0	Port C outputs bit 0
[1]	PORTC1	Port C outputs bit 1

[2]	PORTC2	Port C outputs bit 2
[3]	PORTC3	Port C outputs bit 3
[4]	PORTC4	Port C outputs bit 4
[5]	PORTC5	Port C outputs bit 5
[6]	PORTC6	Port C outputs bit 6
[7]	-	Keep not used

Port **C** direction register **- DDRC**

DDRC - port **C** direction register

DDRC: 0x07 (0x27) Default: 0x00

DDRC	-	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0

R / W	-	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	-	0	0	0	0	0	0	0

[0]	DDC0	PC0 direction control bit
[1]	DDC1	PC1 direction control bit
[2]	DDC2	PC2 direction control bit
[3]	DDC3	PC3 direction control bit
[4]	DDC4	PC4 direction control bit
[5]	DDC5	PC5 direction control bit
[6]	DDC6	PC6 direction control bit
[7]	-	Keep not used

Port **C** Input Data Register **- PINC**

PINC - Port **C** input data register

PINB: 0x06 (0x26) Default: 0x00

PINC	-	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
R / W	-	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	-	0	0	0	0	0	0	0

[0]	PINC0	PC0 port data
[1]	PINC1	PC1 port data
[2]	PINC2	PC2 port data
[3]	PINC3	PC3 port data
[4]	PINC4	PC4 port data
[5]	PINC5	PC5 port data
[6]	PINC6	PC6 port data
[7]	-	Keep not used

Port **D** Output Data Register **- PORTD**

PORTD - Port **D** Output Data Register

PORTD: 0x0B (0x2B) Default: 0x00

PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD1	PORTD2	PORTD1	PORTD0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	0	0	0	0	0	0	0	0

[0]	PORTD0	Port D outputs bit 0
[1]	PORTD1	Port D outputs bit 1
[2]	PORTD2	Port D outputs bit 2
[3]	PORTD3	Port D outputs bit 3
[4]	PORTD4	Port D outputs bit 4
[5]	PORTD5	Port D outputs bit 5
[6]	PORTD6	Port D outputs bit 6
[7]	PORTD7	Port D outputs bit 7

Port **D** Direction Register **- DDRD**

DDRD - port **D** direction register

DDRD: 0x0A (0x2A) Default: 0x00

DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	0	0	0	0	0	0	0	0

[0]	DDD0	PD0 direction control bit
[1]	DDD1	PD1 direction control bit
[2]	DDD2	PD2 direction control bit
[3]	DDD3	PD3 direction control bit
[4]	DDD4	PD4 direction control bit
[5]	DDD5	PD5 direction control bit
[6]	DDD6	PD6 direction control bit
[7]	DDD7	PD7 direction control bit

Port **D** Input Data Register **- PIND**

PIND - Port **D** input data register

PIND: 0x09 (0x29) Default: 0x00

PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W

Initial value	0	0	0	0	0	0	0	0

[0]	PIND0	PD0 port data
[1]	PIND1	PD1 port data
[2]	PIND2	PD2 port data
[3]	PIND3	PD3 port data
[4]	PIND4	PD4 port data
[5]	PIND5	PD5 port data
[6]	PIND6	PD6 port data
[7]	PIND7	PD7 port data

Port **E** Output Data Register **- PORTE**

PORTE - Port **E** output data register

PORTE: 0xA9 Default: 0x00

PORTE	-	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0
R / W	-	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	-	0	0	0	0	0	0	0

[0]	PORTE0	Port E outputs bit 0
[1]	PORTE1	Port E outputs bit 1
[2]	PORTE2	Port E outputs bit 2
[3]	PORTE3	Port E outputs bit 3
[4]	PORTE4	Port E outputs bit 4
[5]	PORTE5	Port E outputs bit 5
[6]	PORTE6	Port E outputs bit 6
[7]	-	Keep not used

Port **E** Direction Register **- DDRE**

DDRE - Port **E** direction register

DDRE: 0xA8 Default: 0x00

DDRE	-	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0
R / W	-	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	-	0	0	0	0	0	0	0

[0]	DDE0	PE0 direction control bit
[1]	DDE1	PE1 direction control bit
[2]	DDE2	PE2 direction control bit
[3]	DDE3	PE3 direction control bit

[4]	DDE4	PE4 direction control bit
[5]	DDE5	PE5 direction control bit
[6]	DDE6	PE6 direction control bit
[7]	-	Keep not used

Port **E** Input Data Register **- PINE**

PINE - Port **E** input data register

PINE: 0xA7 Default: 0x00

PINE	-	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0
R / W	-	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial value	-	0	0	0	0	0	0	0

[0]	PINE0	PE0 port data
[1]	PINE1	PE1 port data
[2]	PINE2	PE2 port data
[3]	PINE3	PE3 port data
[4]	PINE4	PE4 port data
[5]	PINE5	PE5 port data
[6]	PINE6	PE6 port data
[7]	-	Keep not used

Pin level change interrupt

- 30 pin change interrupt sources
- 4 interrupt entries

Summary

The pin change interrupt is triggered by the PBn, PCn, PDn and PEn pins. As long as the pin change interrupt is enabled, ie Making these pins configured as an output can also trigger an interrupt. This can be used to generate software interrupts.

Any enabled PBn pin flip will trigger pin level interrupt PCI0, enable PCn pin flip will trigger PCI1,
The enabled PDn pin flips to trigger PCI2, and the enabled PEn pin flips to trigger PCI3. Each pin is interrupted
The enable is controlled by the PCMSK0, PCMSK1, PCMSK2 and PCMSK3 registers, respectively. All pin levels are changed
Are detected asynchronously and can be used as a wake source in some sleep modes.

Register definition

Pin Change Interrupt register list

register	address	Defaults	description
PCICR	0x68	0x00	The pin changes the interrupt control register
PCIFR	0x3B	0x00	The pin changes the interrupt flag register
PCMSK0	0x6B	0x00	Pin Change Interrupt Mask Register 0
PCMSK1	0x6C	0x00	Pin Change Interrupt Mask Register 1
PCMSK2	0x6D	0x00	Pin Change Interrupt Mask Register 2
PCMSK3	0x73	0x00	Pin Change Interrupt Mask Register 3

PCICR - Pin change interrupt control register

PCICR - Pin change interrupt control register

Address: 0x68 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	PCIE3	PCIE2	PCIE1	PCIE0
R / W	-	-	-	-	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 4	-	Keep it.
3	PCIE3	Pin Change Interrupt Enable Control Bit 3. When the PCIE3 bit is set to "1" and the global interrupt is enabled, pin change interrupt 3 is enabled. Any one to make The PEn pin can change the PCI3 interrupt. The enable of the PEn pin interrupt can be determined by PCMSK3 register to control. When the PCIE3 bit is set to "0", the pin change interrupt 3 is disabled.
2	PCIE2	Pin Change Interrupt Enable Control Bit 2.

Page 76

When the PCIE2 bit is set to "1" and the global interrupt is enabled, pin change interrupt 2 is enabled. Any one to make The PDn pin can change the PCI2 interrupt. The enable of the PDn pin interrupt can be determined by PCMSK2 register to control.
When the PCIE2 bit is set to "0", the pin change interrupt 2 is disabled.

1 PCIE1 Pin Change Interrupt Enable Control bit 1.
When the PCIE1 bit is set to "1" and the global interrupt is enabled, pin change interrupt 1 is enabled. Any one to make The PCn pin can change the PCI1 interrupt. PCn pin interrupt enable can be made by PCMSK1 register to control.
When the PCIE1 bit is set to "0", pin change interrupt 1 is disabled.

0 PCIE0 Pin Change Interrupt Enable Control Bit 0.
When the PCIE0 bit is set to "1" and the global interrupt is enabled, pin change interrupt 0 is enabled. Any one to make Can change the level of the PBn pin will produce PCI0 interrupt. The enable of the PBn pin interrupt can be determined by PCMSK0 register to control.
When the PCIE0 bit is set to "0", the pin change interrupt 0 is disabled.

PCIFR - Pin Change Interrupt Flag Register

PCIFR - Pin Change Interrupt Flag Register

Address: 0x3B Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	PCIF3	PCIF2	PCIF1	PCIF0
R / W	-	-	-	-	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 4	-	Keep it.
3	PCIF3	Pin change interrupt flag bit 3. The PCIV3 is set for any of the enabled PEn pin changes. When both PCIE3 and global interrupts are set Bit, the MCU will jump to the PCI3 interrupt entry address. The enable of the PEn pin interrupt can be selected by PCMSK3 Register to control. The execution of the interrupt service routine or writing "1" to the PCIF3 bit will clear the PCIF3 bit.
2	PCIF2	Pin change interrupt flag bit 2. The level change of any enabled PDn pin will set PCIF2. When both PCIE2 and global interrupts are set Bit, the MCU will jump to the PCI2 interrupt entry address. The PDn pin interrupt enable is enabled by PCMSK2 Register to control. The execution of the interrupt service routine or writing "1" to the PCIF2 bit will clear the PCIF2 bit.
1	PCIF1	Pin change interrupt flag bit 1. The level change of any enabled PCn pin sets PCIF1. When both PCIE1 and global interrupts are set Bit, the MCU will jump to the PCI1 interrupt entry address. The PCn pin interrupt can be enabled by PCMSK1, respectively Register to control. The execution of the interrupt service routine or writing "1" to the PCIF1 bit clears the PCIF1 bit.
0	PCIF0	The pin changes the interrupt flag bit 0. The level change of any enabled PBn pin sets PCIF0. When both PCIE0 and global interrupts are set

Page 77

Bit, the MCU will jump to the PCI0 interrupt entry address. The enable of the PBn pin interrupt can be set by PCMSK0

Register to control.

The execution of the interrupt service routine or writing "1" to the PCIF0 bit clears the PCIF0 bit.

PCMSK0 - Pin Change Interrupt Mask Register 0

PCMSK0 - Pin Change Mask Register 0

Address: 0x6B Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	PCINT7	Pin change enable mask 7. When the PCINT7 bit is set to "1", the PB7 pin change interrupt is enabled. The level on the PB7 pin The change will set PCIF0. If PCIE0 bit and global interrupt are set, a PCI0 interrupt will be generated. When the PCINT7 bit is set to "0", the PB7 pin change interrupt is disabled.
6	PCINT6	Pin change enable mask bit 6. When the PCINT6 bit is set to "1", the PB6 pin change interrupt is enabled. The level on the PB6 pin The change will set PCIF0. If PCIE0 bit and global interrupt are set, a PCI0 interrupt will be generated. When setting the PCINT6 bit to "0", the PB6 pin change interrupt is disabled.
5	PCINT5	Pin change enable mask 5. When the PCINT5 bit is set to "1", the PB5 pin change interrupt is enabled. The level on the PB5 pin The change will set PCIF0. If PCIE0 bit and global interrupt are set, a PCI0 interrupt will be generated. When setting the PCINT5 bit to "0", the PB5 pin change interrupt is disabled.
4	PCINT4	Pin change enable mask bit 4. When the PCINT4 bit is set to "1", the PB4 pin change interrupt is enabled. The level on the PB4 pin The change will set PCIF0. If PCIE0 bit and global interrupt are set, a PCI0 interrupt will be generated. When the PCINT4 bit is set to "0", the PB4 pin change interrupt is disabled.
3	PCINT3	Pin change enable mask bit 3. When the PCINT3 bit is set to "1", the PB3 pin change interrupt is enabled. The level on the PB3 pin The change will set PCIF0. If PCIE0 bit and global interrupt are set, a PCI0 interrupt will be generated. When the PCINT3 bit is set to "0", the PB3 pin change interrupt is disabled.
2	PCINT2	Pin change enable mask bit 2. When the PCINT2 bit is set to "1", the PB2 pin change interrupt is enabled. The level on the PB2 pin The change will set PCIF0. If PCIE0 bit and global interrupt are set, a PCI0 interrupt will be generated. When setting the PCINT2 bit to "0", the PB2 pin change interrupt is disabled.
1	PCINT1	Pin change enable mask bit 1. When the PCINT1 bit is set to "1", the PB1 pin change interrupt is enabled. The level on the PB1 pin The change will set PCIF0. If PCIE0 bit and global interrupt are set, a PCI0 interrupt will be generated. When setting the PCINT1 bit to "0", the PB1 pin change interrupt is disabled.

0	PCINT0	Pin change enable mask bit 0. When setting the PCINT0 bit to "1", the PB0 pin change interrupt is enabled. The level on the PB0 pin The change will set PCIF0. If PCIE0 bit and global interrupt are set, a PCI0 interrupt will be generated. When setting the PCINT0 bit to "0", the PB0 pin change interrupt is disabled.

PCMSK1 - Pin Change Interrupt Mask Register 1

PCMSK1 - Pin Change Mask Register 1

Address: 0x6C Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
R / W	-	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	-	Keep it.
6	PCINT14	pin change enable mask 14. When the PCINT14 bit is set to "1", the PC6 pin change interrupt is enabled. The level on the PC6 pin The change will set PCIF1. If PCIE1 bit and global interrupt are set, a PCI1 interrupt will be generated. When the PCINT14 bit is set to "0", the PC6 pin change interrupt is disabled.
5	PCINT13	pin change enable mask bit 13. When the PCINT13 bit is set to "1", the PC5 pin change interrupt is enabled. The level on the PC5 pin The change will set PCIF1. If PCIE1 bit and global interrupt are set, a PCI1 interrupt will be generated. When the PCINT13 bit is set to "0", the PC5 pin change interrupt is disabled.
4	PCINT12	pin change enable mask 12. When the PCINT12 bit is set to "1", the PC4 pin change interrupt is enabled. The level on the PC4 pin The change will set PCIF1. If PCIE1 bit and global interrupt are set, a PCI1 interrupt will be generated. When the PCINT12 bit is set to "0", the PC4 pin change interrupt is disabled.
3	PCINT11	pin change enable mask bit 11. When the PCINT11 bit is set to "1", the PC3 pin change interrupt is enabled. The level on the PC3 pin The change will set PCIF1. If PCIE1 bit and global interrupt are set, a PCI1 interrupt will be generated. When the PCINT11 bit is set to "0", the PC3 pin change interrupt is disabled.
2	PCINT10	pin change enable mask bit 2. When the PCINT10 bit is set to "1", the PC2 pin change interrupt is enabled. The level on the PC2 pin The change will set PCIF1. If PCIE1 bit and global interrupt are set, a PCI1 interrupt will be generated. When the PCINT10 bit is set to "0", the PC2 pin change interrupt is disabled.
1	PCINT9	Pin change enable mask bit 1. When the PCINT9 bit is set to "1", the PC1 pin change interrupt is enabled. The level on the PC1 pin The change will set PCIF1. If PCIE1 bit and global interrupt are set, a PCI1 interrupt will be generated. When the PCINT9 bit is set to "0", the PC1 pin change interrupt is disabled.
0	PCINT8	Pin change enable mask bit 0. When the PCINT8 bit is set to "1", the PC0 pin change interrupt is enabled. The level on the PC0 pin The change will set PCIF1. If PCIE1 bit and global interrupt are set, a PCI1 interrupt will be generated. When setting the PCINT8 bit to "0", the PC0 pin change interrupt is disabled.

PCMSK2 - Pin Change Interrupt Mask Register **2**

PCMSK2 **-** Pin Change Mask Register **2**

Address: 0x6D Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	PCINT23	pin change enable mask bit 23. When the PCINT23 bit is set to "1", the PD7 pin change interrupt is enabled. PD7 pin on the power The ping change will set PCIF2. If PCIE2 bit and global interrupt are set, a PCI2 interrupt will be generated. When the PCINT23 bit is set to "0", the PD7 pin change interrupt is disabled.
6	PCINT22	pin change enable mask bit 6.

When the PCINT22 bit is set to "1", the PD6 pin change interrupt is enabled. PD6 pin on the power

The ping change will set PCIF2. If PCIE2 bit and global interrupt are set, a PCI2 interrupt will be generated.

When the PCINT22 bit is set to "0", the PD6 pin change interrupt is disabled.

5 PCINT21 pin change enable mask bit 21.

When the PCINT21 bit is set to "1", the PD5 pin change interrupt is enabled. PD5 pin on the power

The ping change will set PCIF2. If PCIE2 bit and global interrupt are set, a PCI2 interrupt will be generated.

When the PCINT21 bit is set to "0", the PD5 pin change interrupt is disabled.

4 PCINT20 pin change enable mask bit 20.

When the PCINT20 bit is set to "1", the PD4 pin change interrupt is enabled. PD4 pin on the power

The ping change will set PCIF2. If PCIE2 bit and global interrupt are set, a PCI2 interrupt will be generated.

When the PCINT20 bit is set to "0", the PD4 pin change interrupt is disabled.

3 PCINT19 pin change enable mask bit 19.

When the PCINT19 bit is set to "1", the PD3 pin change interrupt is enabled. PD3 pin on the power

The ping change will set PCIF2. If PCIE2 bit and global interrupt are set, a PCI2 interrupt will be generated.

When the PCINT19 bit is set to "0", the PD3 pin change interrupt is disabled.

2 PCINT18 pin change enable mask 18.

When the PCINT18 bit is set to "1", the PD2 pin change interrupt is enabled. PD2 pin on the power

The ping change will set PCIF2. If PCIE2 bit and global interrupt are set, a PCI2 interrupt will be generated.

When the PCINT18 bit is set to "0", the PD2 pin change interrupt is disabled.

1 PCINT17 pin change enable mask bit 17.

When the PCINT17 bit is set to "1", the PD1 pin change interrupt is enabled. PD1 pin on the power

The ping change will set PCIF2. If PCIE2 bit and global interrupt are set, a PCI2 interrupt will be generated.

When the PCINT17 bit is set to "0", the PD1 pin change interrupt is disabled.

0 PCINT16 pin change enable mask bit 16.

When the PCINT16 bit is set to "1", the PD0 pin change interrupt is enabled. PD0 pin on the power

The ping change will set PCIF2. If PCIE2 bit and global interrupt are set, a PCI2 interrupt will be generated.

When the PCINT16 bit is set to "0", the PD0 pin change interrupt is disabled.

PCMSK3 - Pin Change Interrupt Mask Register **3**

PCMSK3 **-** Pin Change Mask Register **3**

Address: 0x73 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	PCINT30	PCINT29	PCINT28	PCINT27	PCINT26	PCINT25	PCINT24
R / W	-	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit Name description

7 - Keep it.

6 PCINT30 pin change enable mask bit 30.

When the PCINT30 bit is set to "1", the PE6 pin change interrupt is enabled. PE6 pin on the power

Flat change will set PCIF3. If PCIE3 bit and global interrupt are set, PCI3 interrupt will be generated.

When the PCINT30 bit is set to "0", the PE6 pin change interrupt is disabled.

5 PCINT29 pin change enable mask bit 39.

When the PCINT29 bit is set to "1", the PE5 pin change interrupt is enabled. PE5 pin on the power

Flat change will set PCIF3. If PCIE3 bit and global interrupt are set, PCI3 interrupt will be generated.

When the PCINT29 bit is set to "0", the PE5 pin change interrupt is disabled.

4 PCINT28 pin change enable mask bit 28.

When the PCINT28 bit is set to "1", the PE4 pin change interrupt is enabled. PE4 pin on the power

Flat change will set PCIF3. If PCIE3 bit and global interrupt are set, PCI3 interrupt will be generated.

When the PCINT28 bit is set to "0", the PE4 pin change interrupt is disabled.

3	PCINT27 pin change enable mask bit 27. When the PCINT27 bit is set to "1", the PE3 pin change interrupt is enabled. PE3 pin on the power Flat change will set PCIF3. If PCIE3 bit and global interrupt are set, PCI3 interrupt will be generated. When the PCINT27 bit is set to "0", the PE3 pin change interrupt is disabled.
2	PCINT26 pin change enable mask bit 26. When the PCINT26 bit is set to "1", the PE2 pin change interrupt is enabled. PE2 pin on the power Flat change will set PCIF3. If PCIE3 bit and global interrupt are set, PCI3 interrupt will be generated. When the PCINT26 bit is set to "0", the PE2 pin change interrupt is disabled.
1	PCINT25 pin change enable mask bit 25. When the PCINT25 bit is set to "1", the PE1 pin change interrupt is enabled. PE1 pin on the power Flat change will set PCIF3. If PCIE3 bit and global interrupt are set, PCI3 interrupt will be generated. When setting the PCINT25 bit to "0", the PE1 pin change interrupt is disabled.
0	PCINT24 pin change enable mask bit 24. When the PCINT24 bit is set to "1", the PE0 pin change interrupt is enabled. PE0 pin on the power Flat change will set PCIF3. If PCIE3 bit and global interrupt are set, PCI3 interrupt will be generated. When setting PCINT24 bit to "0", the PE0 pin change interrupt is disabled.

8 -bit timer / counter *0*

- 8-bit counter
- Two separate comparison units
- When the compare match occurs, the counter is automatically cleared and loaded automatically
- Phase-corrected PWM output without interference pulse
- Frequency generator
- External event counter
- 10-bit clock prescaler
- overflow and compare match interrupt
- with dead time control
- 8 selectable trigger sources automatically turn off the PWM output

High-speed, high-resolution (500KHz @ 7Bit) PWM in high-speed clock mode

Overview

TC0 is a general-purpose 8-bit timer counter module that supports PWM output and can produce waveforms precisely. TC0 package
With one count clock generation unit, one 8-bit counter, waveform generation mode control unit and two output compare single
yuan. At the same time, TC0 can be shared with TC1 10-bit prescaler, you can also use the 10-bit prescaler. Pre-divided
Frequency on the system clock clkio or high speed clock rcm2x (internal 32M RC oscillator output clock rc32m 2 times the frequency)
Divide to produce the count clock Clkt0. The waveform generation mode control unit controls the operating mode and comparison of the counter
Out of the waveform. Depending on the operating mode, the counter is cleared for each count clock Clkt0, plus one or
Minus one operation. Clkt0 can be generated by an internal clock source or an external clock source. When the counter count value TCNT0 reaches the maximum
The value (equal to the maximum value 0xFF or the output compare register OCR0A, defined as TOP, defines the maximum value of MAX to show the area
Other), the counter will be cleared or decremented. When the counter count value TCNT0 reaches the minimum value (equal to
0x00, defined as BOTTOM), the counter will be added to an operation. When the counter count value TCNT0 arrives
OCR0A / OCR0B, also known as a compare match, will clear or set the output compare signal OC0A / OC0B, to
Generates PWM waveforms. When the dead time is enabled, the dead time set (the count corresponding to the DTR0 register
Clock) will be inserted into the generated PWM waveform. The software can be turned off by clearing the COM0A / COM0B bit to zero
OC0A / OC0B waveform output, or set the corresponding trigger source, when the trigger event occurs when the hardware automatically cleared
COM0A / COM0B bits to turn off the OC0A / OC0B waveform output.

The count clock can be generated by an internal or external clock source. The clock source selection and frequency selection are determined by the TCCR0B register
CS0 bits to control, see the TC0 and TC1 prescaler sections for details.

The length of the counter is 8 bits and supports bidirectional counting. The waveform generation mode is the operating mode of the counter located by TCCR0A and
The WGM0 bit of the TCCR0B register is controlled. According to the different operating modes, the counter for each count clock Clkt0
To achieve zero, plus one or minus one operation. When the count overflows, the count overflow flag TOV0 is located in the TIFR0 register
Bit will be set. TC0 count overflow interrupt can be generated when interrupt is enabled.

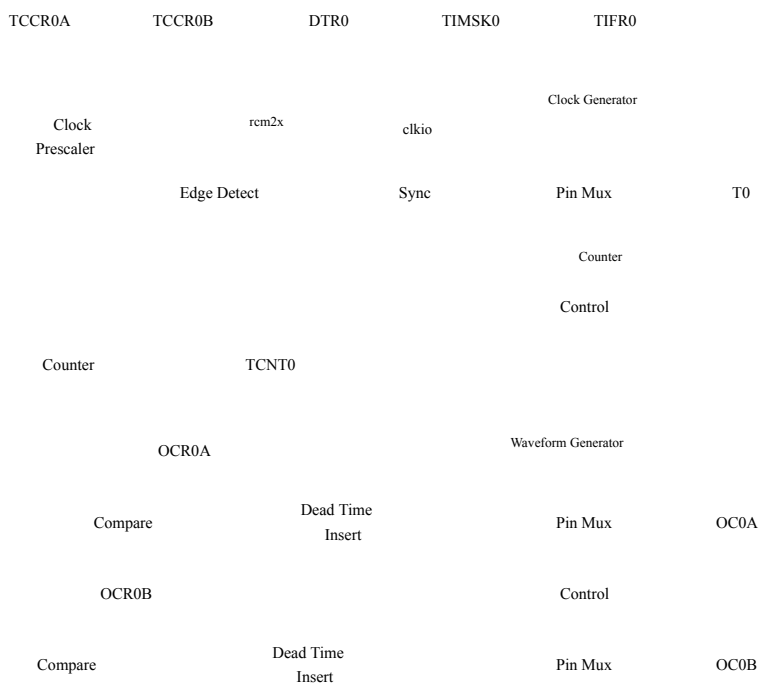
The output compare unit compares the count value TCNT0 with the values of the output compare registers OCR0A and OCR0B. When TCNT0
Equal to OCR0A or OCR0B is called a compare match, the output compare flag OCF0A in the TIFR0 register or
The OCF0B bit is set. TC0 output compare match interrupt can be generated when interrupt is enabled.
Note that in the PWM operating mode, the OCR0A and OCR0B registers are double buffered registers. In normal mode
And CTC mode, the double buffering function is disabled. When the count reaches the maximum or minimum value, the value in the buffer register is synchronized

Page 82

New to compare registers OCR0A and OCR0B. See the description of the working mode section.

The waveform generator generates and controls the output mode control according to the waveform generation mode and uses the compare match and the count overflow to generate
Output compare waveform signals OC0A and OC0B. The specific way of generating is described in the working mode and register section description. To output
When comparing the waveform signals OC0A and OC0B to the corresponding pins, the data direction register for this pin must also be set to
Output.

The picture below shows the internal structure of TC0. TC0 contains one count clock generation unit, one 8-bit counter, two inputs
A comparison unit and two waveform generation control units.



TC0 structure diagram

Operating mode

Timing counter 0 has four different modes of operation, including normal mode (Normal), compare match clear (CTC)
Mode, fast pulse width modulation (FPWM) mode and phase correction pulse width modulation (PCPWM) mode,
Generates the mode control bit WGM0 [2: 0] to select. The four modes are described below in detail. Because there are two independent output ratios
The units are represented by "A" and "B", respectively, and the lower output "x" is used to represent the two output compare cell channels.

Normal mode

The normal mode is the simplest mode of operation for the timer counter. The waveform generation mode control bit, WGM0 [2: 0] = 0,

The maximum value of the TOP is MAX (0xFF). In this mode, the count mode is incremented by one for each count clock
After the counter reaches the TOP overflow, it returns to BOTTOM to restart. In the same value as the count value TCNT0 becomes zero
Set the timer counter overflow flag TOV0 in the clock. The TOV0 flag in this mode is like the 9th count bit, just
Will only be set to not be cleared. The overflow interrupt service routine automatically clears the TOV0 flag, which the software can use to improve it
The resolution of the counter. There is no special case in normal mode to consider, you can always write a new count value.
The waveform of the output compare signal OC0x is obtained by setting the data direction register of the OC0x pin to output. When COM0x = 1
, The OC0x signal is flipped when a compare match occurs. The frequency of the waveform in this case can be calculated using the following formula:

$$f_{oc0xnormal} = f_{sys} / (2 * N * 256)$$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).
The output compare unit can be used to generate interrupts, but interrupts are not recommended in normal mode, which takes too much CPU
time.

CTC mode

When the WGM0 [2: 0] = 2 is set, the timer counter 0 enters the CTC mode and the maximum value of the count is OCR0A. in
In this mode, the count mode is incremented by one for each count clock. When the value of the counter TCNT0 equals TOP
The counter is cleared. OCR0A defines the maximum value of the count, that is, the resolution of the counter. This mode allows the user to be tolerant
Easy control matches the frequency of the output and also simplifies the operation of the external event count.
When the counter reaches the maximum value of the count, the output compare match flag OCF0 is set and the corresponding interrupt enable is set
Will be interrupted. The OCR0A register can be updated in the interrupt service routine to count the maximum value. In this mode
OCR0A does not use double buffering, the maximum value of the counter in the absence of a prescaler or a very low prescaler operation is updated to
Be careful when approaching the minimum. If the value written to OCR0A is less than the value of TCNT0 at that time, the counter will be lost
One match match. Before the next match match occurs, the counter has to count to TOP before starting from BOTTOM
Start counting to OCR0A value. As with the normal mode, the count value is set back to the BOTTOM count clock to set the TOV0 flag
Mind
The waveform of the output compare signal OC0x is obtained by setting the data direction register of the OC0x pin to output. When COM0x = 1
, The OC0x signal is flipped when a compare match occurs. The frequency of the waveform in this case can be calculated using the following formula:

$$f_{oc0xcte} = f_{sys} / (2 * N * (1 + OCR0x))$$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).
As can be seen from the formula, when the set OCR0A is 0x0 and no prescaler, you can get the maximum frequency of f_{sys} / 2 output
Waveform.

Fast PWM mode

When setting WGM0 [2: 0] = 3 or 7, the timer counter 0 enters the fast PWM mode and can be used to generate high frequency
PWM waveform, the maximum value TOP is MAX (0xFF) or OCR0x, respectively. Fast PWM mode and other PWM modes
The difference is that it is a one-way operation. The counter is incremented from the minimum value 0x00 to TOP and then back to BOTTOM to count.
When the count value TCNT0 reaches OCR0x or BOTTOM, the output compare signal OC0x is set or cleared, depending on
Compare the output mode COM0x settings, see the register description for details. Due to the use of one-way operation, fast PWM mode
The operating frequency is twice the phase corrected PWM mode using bidirectional operation. High frequency characteristics make fast PWM mode available
In power regulation, rectification and DAC applications. High-frequency signals can be reduced by the size of external components (inductive capacitors, etc.)
While reducing system costs.
When the count value reaches the maximum value, the timer counter overflow flag TOV0 will be set and the compare buffer value will be updated
To the comparison value. If the interrupt is enabled, the compare buffer OCR0x register can be updated in the interrupt service routine.
The waveform of the output compare signal OC0x is obtained by setting the data direction register of the OC0x pin to output. The frequency of the waveform

The rate can be calculated using the following formula:

$f_{oc0xfpwm} = f_{sys} / (N * (1 + TOP))$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).

When a compare match occurs between TCNT0 and OCR0x, the waveform generator sets (clears) the OC0x signal. When TCNT0 is cleared

At zero, the waveform generator clears (sets) the OC0x signal to generate a PWM wave. Whereby the extreme value of OCR0x will be

Will produce a special PWM waveform. When OCR0x is set to 0x00, the output PWM is every (1 + TOP) count clock

There is a narrow spike. When OCR0x is set to the maximum value, the output waveform is a continuous high or low level.

Phase correction PWM mode

When setting WGM0 [2: 0] = 1 or 5, the timer counter 0 enters the phase corrected PWM mode, the maximum value of the count

TOP is MAX (0xFF) or OCR0A, respectively. The counter is bidirectional, incremented by BOTTOM to TOP, and then again

Decrements to BOTTOM, and repeats this operation. When the count reaches TOP and BOTTOM, the counting direction is changed and the count value is

TOP or BOTTOM only stay on a count clock. In increments or decrements, the count values TCNT0 and OCR0x

When matching, the output compare signal OC0x will be cleared or set, depending on the setting of the compare output mode COM0x. With a single

Compared to the operation, the maximum frequency available for bidirectional operation is small, but its excellent symmetry is more suitable for motor control.

In phase correction PWM mode, the TOV0 flag is set when the count reaches BOTTOM. When the count reaches TOP, compare

The value of the buffer is updated to the comparison value. If the interrupt is enabled, the compare buffer OCR0x can be updated in the interrupt service routine

Deposit.

The waveform of the output compare signal OC0x is obtained by setting the data direction register of the OC0x pin to output. The frequency of the waveform

The rate can be calculated using the following formula:

$f_{oc0xpcpwm} = f_{sys} / (N * TOP * 2)$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).

During waveform counting, when the TCNT0 matches OCR0x, the waveform generator clears (sets) the OC0x signal. in

In the process of decrementing the count, when the TCNT0 matches OCR0x, the waveform generator sets (clears) the OC0x signal. thus

The extreme value of OCR0x produces a special PWM wave. When OCR0x is set to the maximum or minimum value, OC0x signal is output

Will remain low or high.

In order to ensure that the output PWM wave symmetry on both sides of the minimum value, in the absence of a comparison match, there are two cases

Flip OC0x signal. The first case is when the value of OCR0x changes from the maximum value 0xFF to other data. When OCR0x

Is the maximum value, the count value reaches the maximum, the output of OC0x is the same as that of the previous descending order.

Hold OC0x unchanged. The value of the new OCR0x will be updated (non-0xFF), the value of OC0x will remain

When the ascending order counts, a compare match occurs. At this point OC0x signal is not symmetrical with the minimum value, so need to

TCNT0 to reach the maximum value when the flip OC0x signal, which does not occur when the comparison match the OC0x signal flip the first situation

condition. The second case is that when TCNT0 starts counting from a value higher than OCR0x, it will lose a compare match,

Thus causing the occurrence of asymmetric situations. Also need to flip OC0x signal to achieve the minimum on both sides of the symmetry.

Dead time control

When the DTEN0 bit is set to "1", the function of inserting the dead time is enabled and the output waveform of OC0A and OC0B will be set to B

Channel comparison output generated by the waveform based on the insertion of the set dead time, the length of the time for the DTR0 register count

The number of clocks corresponding to the time value. As shown in the following figure, the dead time insertion of OC0A and OC0B is the comparison of channel B

Output waveform as a reference. When COM0A and COM0B are the same as "2" or "3", OC0A's waveform polarity is the same as OC0B

Waveforms are the same, when COM0A and COM0B are "2" or "3", OC0A waveform and OC0B waveform

The opposite polarity.

Bottom OCR0B TOP OCR0B TOP

Figure 1 TC0 dead time control in FPWM mode

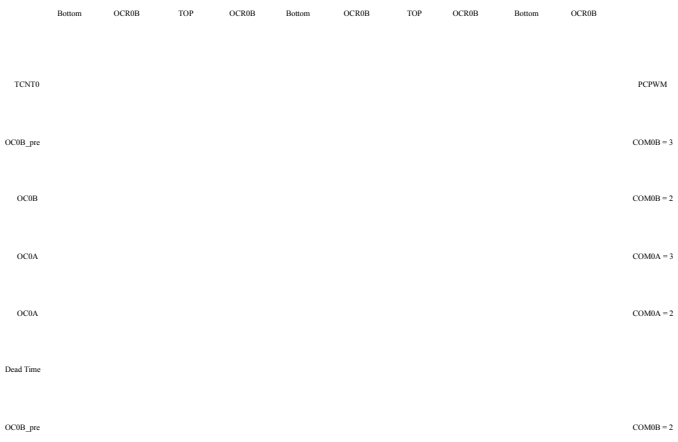


Figure 2 TC0 dead time control in PCPWM mode

When the DTEN0 bit is set to "0", the function of inserting the dead time is disabled. The output waveforms of OC0A and OC0B are the respective More than the output generated by the waveform.

High speed clock mode

In high-speed clock mode, a higher frequency clock is used as the clock source for counting to produce higher speed and higher resolution Rate of the PWM waveform. This high-frequency clock is multiplied by the output clock rc32m of the internal 32M RC oscillator produced. Therefore, before entering the high-frequency mode, you must first enable the internal 32M RC oscillator multiplier function, that is set TCKCSR register of the F2XEN bit and wait for a certain period of time until the multiplier clock signal output is stable. Then, it can be set The TC2XS0 bit of TCKCSR enables the timer counter to enter high-speed clock mode.

In this mode, the system clock is asynchronous to the high-speed clock, and some registers (see the TC0 register list) are operating In the high-speed clock domain, therefore, when configuring and reading such registers is also asynchronous, the operation should pay attention.

There is no special requirement for non-continuous read / write operation of the register in the high-speed clock domain, and when a continuous read / write operation is performed, To wait for a system clock, follow these steps:

1) write register A;
2) wait for a system clock (NOP or operating system clock under the register);
3) Read or write to register A or B.
4) Wait for a system clock (NOP or operating system clock under register).

When reading a register in a high-speed clock domain, the register other than TCNT0 can be read directly. When the counter When counting is in progress, the value of TCNT0 changes with the high-speed clock, and the counter can be paused (set CS0 to zero) and then read The value of TCNT0.

Register definition

TC0 register list

register	address	Defaults	description
TCCR0A *	0x44	0x00	TC0 control register A
TCCR0B *	0x45	0x00	TC0 control register B
TCNT0 *	0x46	0x00	TC0 count value register
OCR0A *	0x47	0x00	TC0 output compare register A
OCR0B *	0x48	0x00	TC0 output compare register B
DSX0 *	0x49	0x00	TC0 Trigger Source Control Register
DTR0 *	0x4F	0x00	TC0 dead time register
TIMSK0	0x6E	0x00	Timer counter 0 interrupt mask register
TIFR0	0x35	0x00	Timer counter 0 interrupt flag register
TCKCSR	0xEC	0x00	TC clock control and status register

【note】

The register with "*" operates on the system clock and the high-speed clock domain. When the register with "*" only works on the system Under the clock domain.

TC0 control register **A-TCCR0A**

TCCR0A - TC0 Control Register A

Address: 0x44 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	COM0A1	COM0A0	COM0B1	COM0B0	DOC0B	DOC0A	WGM01	WGM00
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	COM0A1	TC0 Compare Match A Output Mode Control High. COM0A1 and COM0A0 together constitute the comparison output mode control COM0A [1: 0], used to control OC0A output waveform. If bit 1 or bit 2 of COM0A is set, the output compare waveform is occupied According to the OC0A pin, but the pin of the data direction register must be set to output this waveform. In different modes of operation, COM0A on the output comparison waveform control is also different, see the specific comparison Out mode control table description.
6	COM0A0	TC0 compare match A output mode control low. COM0A0 and COM0A1 together make up compare output mode control COM0A [1: 0], used to control OC0A output waveform. If bit 1 or bit 2 of COM0A is set, the output compare waveform is occupied According to the OC0A pin, but the pin of the data direction register must be set to output this waveform. In different modes of operation, COM0A on the output comparison waveform control is also different, see the specific comparison Out mode control table description.
5	COM0B1	TC0 compare match B output mode control high. COM0B1 and COM0B0 together make up compare output mode control COM0B [1: 0], used to control OC0B

Page 88

4	COM0B0	Of the output waveform. If the 1 or 2 bits of COM0B are set, the output compare waveform is occupied OC0B pin, but the pin's data direction register must be set high to output this waveform. No With the work mode, COM0B on the output comparison waveform control is also different, see the specific comparison output mode Type control table description. TC0 compare match B output mode control low. COM0B0 and COM0B1 together constitute the compare output mode control COM0B [1: 0], used to control OC0B output waveform. If bit 1 or 2 of COM0B is set, the output compare waveform is occupied With the OC0B pin, but the pin's data direction register must be set high to output this waveform. in Different operating modes, COM0B on the output comparison waveform control is also different, see the specific comparison output Mode control table description.
3	DOC0B	TC0 OFF Output Compare Enable Control High. When the DOC0B bit is set to "1", the trigger source off output compare signal OC0B is enabled. When made When the event is triggered, the hardware automatically turns off the OC0B's waveform output. When the set value of the DOC0B bit is set to "0", the trigger source off output compare signal OC0B is disabled. When made When the event is triggered, the OC0B's waveform output is not turned off.
2	DOC0A	TC0 OFF Output Compare Enable Control Low. When the DOC0A bit is set to "1", the trigger source turns off the output compare signal OC0A is enabled. When made When the event is triggered, the hardware automatically turns off the OC0A waveform output. When the set value of the DOC0A bit is set to "0", the trigger source off output compare signal OC0A is disabled. When made When the event is triggered, the OC0A waveform output is not turned off.
1	WGM01	TC0 waveform generation mode control center. WGM01 and WGM00, WGM02 together form the waveform generation mode control WGM0 [2: 0], control Counter count mode and waveform generation mode, see the waveform generation mode table description.
0	WGM00	TC0 waveform generation mode control low. WGM00 and WGM01, WGM02 together form the waveform generation mode control WGM0 [2: 0], control Counter count mode and waveform generation mode, see the waveform generation mode table description.

TC0 control register **B- TCCR0B**

TCCR0B **-** TC0 control register B

Address: 0x45 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	FOC0A	FOC0B	OC0AS	DTEN0	WGM02	CS02	CS01	CS00
R / W	W	W	W / R	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	FOC0A	TC0 Force Output Compare A control bit. When operating in non-PWM mode, the "1" can be written to the forced output compare bit FOC0A. Type to produce a comparison match. Forcing the match does not set the OCF0A flag, nor will it be overloaded or Clear the timer, but the output pin OC0A will be updated according to the COM0A settings, Just like a real match. The return value of reading FOC0A is always zero.
6	FOC0B	TC0 Force Output Compare B control bits.

Page 89

5	OC0AS	When operating in non-PWM mode, the "1" can be written to the forced output compare bit FOC0B. Type to produce a comparison match. Forcing the match does not set the OCF0B flag, nor will it be overloaded or Clear the timer, but the output pin OC0B will be updated according to COM0B settings, Just like a real match. The return value of reading FOC0B is always zero. OC0A output port selection control bit. When the OC0AS bit is set to "0", the waveform of OC0A is changed from Pin PD6 output; when the OC0AS bit is set to "1", the waveform of OC0A is shifted from pin PE4 Out (QFP32 package is valid).
4	DTEN0	TC0 dead time enable control bit. When the DTEN0 bit is set to "1", the dead time is enabled. OC0A and OC0B are in the B-pass The track is compared to the output generated by the waveform based on the insertion dead time, and the inserted dead time interval is determined by The count time corresponding to the DTR0 register is determined. The OC0A output waveform is polarized by COM0 and COM0B corresponds to the decision, see OC0A insert dead time after the waveform polarity table. When the DTEN0 bit is set to "0", the dead time insertion is disabled and the waveforms of OC0A and OC0B are Each compares the output generated by the waveform.
3	WGM02	TC0 waveform generation mode control high. WGM02 and WGM00, WGM01 together form the waveform generation mode control WGM0 [2: 0], control The counting method and the waveform generation mode of the counter are described in detail, and the waveform generation mode table is described in detail.
2	CS02	TC0 clock selection control high. Used to select the clock source for timer counter 0.
1	CS01	TC0 clock selection control bit. Used to select the clock source for timer counter 0.
0	CS00	TC0 clock selection control low. Used to select the clock source for timer counter 0.

CS0 [2: 0]	description
0	No clock source, stop counting
1	clk $_{sys}$
2	clk $_{sys}$ / 8, from prescaler
3	clk $_{sys}$ / 64, from prescaler
4	clk $_{sys}$ / 256, from prescaler
5	clk $_{sys}$ / 1024, from prescaler
6	External clock T0 pin, falling edge trigger
7	External clock T0 pin, rising edge triggered

The following table shows the control of the output compare waveform for the non-PWM mode (ie, normal mode and CTC mode) system.

COM0x [1: 0]	description
0	OC0x disconnect, general-purpose IO port operation

1	Match the OC0x signal when comparing the match
2	The OC0x signal is cleared when compare match
3	The OC0x signal is set when compare match

The following table shows the control of the output compare waveform for the compare output mode in fast PWM mode.

COM0x [1: 0]	description
0	OC0x disconnect, general-purpose IO port operation
1	Keep it
2	The OC0x signal is cleared when the compare match is set and the OC0x signal is set when the maximum match is made
3	The OC0x signal is set when the compare match is cleared and the OC0x signal is cleared when the maximum match is reached

The following table shows the control of the output compare waveform for the compare output mode in phase correction mode.

COM0x [1: 0]	description
0	OC0x disconnect, general-purpose IO port operation
1	Keep it
2	The OC0x signal is cleared when the compare match is cleared in the ascending count and set at the compare match OC0x signal
3	The OC0x signal is set when the compare match is set in the ascending count. When the compare match is cleared OC0x signal

The following table shows the waveform generation mode control.

WGM0 [2: 0]	Operating mode	TOP value	Update OCR0X moments	Set TOV0 at all times
0	Normal	0xFF	immediately	MAX
1	PCPWM	0xFF	TOP	BOTTOM
2	CTC	OCR0A	immediately	MAX
3	FPWM	0xFF	TOP	MAX
4	Keep it	-	-	-
5	PCPWM	OCR0A	TOP	BOTTOM
6	Keep it	-	-	-
7	FPWM	OCR0A	TOP	TOP

DSX0 - TC0 trigger source control register

DSX0 - *TC0* trigger source control register

Address: 0x49 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	DSX07	DSX06	DSX05	DSX04	DSX03	DSX02	DSX01	DSX00
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	DSX07	TC0 trigger source selection control enabled bit 7. When the DSX07 bit is set to "1", TC1 overflows as the output signal of the compare output waveform OC0A / OC0B The trigger source is enabled. When the DOC0A / DOC0B bit is "1", the interrupt flag of the selected trigger source is sent The rising edge of the register bit will automatically turn off the OC0A / OC0B waveform output. When the DSX07 bit is set to "0", TC1 overflows as the output signal of the compare output waveform OC0A / OC0B The trigger source is disabled.
6	DSX06	TC0 Trigger Source Select Control Enable bit 6.

Page 91

When the DSX06 bit is set to "1", TC2 overflows as the output signal for the compare output waveform OC0A / OC0B
The trigger source is enabled. When the DOC0A / DOC0B bit is "1", the interrupt flag of the selected trigger source is sent
The rising edge of the register bit will automatically turn off the OC0A / OC0B waveform output.
When setting the DSX06 bit to "0", TC2 overflows as the output signal of the compare output waveform OC0A / OC0B
The trigger source is disabled.

5	DSX05	TC0 Trigger Source Select Control Enable Bit 5. When the DSX05 bit is set to "1", the pin level changes by 0 as the output signal The trigger source for OC0A / OC0B is enabled. When the DOC0A / DOC0B bit is "1", the selected trigger source The rising edge of the interrupt flag register bit will automatically turn off the waveform output of OC0A / OC0B. When the DSX05 bit is set to "0", the pin level changes by 0 as the output signal The trigger source for OC0A / OC0B is disabled.
4	DSX04	TC0 trigger source selection control enabled bit 4. When the DSX04 bit is set to "1", the external interrupt 0 is used to turn off the output compare signal waveform The trigger source for OC0A / OC0B is enabled. When the DOC0A / DOC0B bit is "1", the selected trigger source The rising edge of the interrupt flag register bit will automatically turn off the waveform output of OC0A / OC0B. When the DSX04 bit is set to "0", the external interrupt 0 is used to turn off the output compare signal waveform The trigger source for OC0A / OC0B is disabled.
3	DSX03	TC0 Trigger Source Select Control Enable Bit 3. When setting the DSX03 bit to "1", the analog comparator 1 is shipped with channel 1 as the output compare The trigger source for signal waveform OC0A / OC0B is enabled. When the DOC0A / DOC0B bit is "1" The rising edge of the interrupt flag register bit will automatically turn off the waveform of OC0A / OC0B. Out. When the DSX03 bit is set to "0", the analog comparator 1 is shipped with channel 1 as the output compare The trigger source for signal waveform OC0A / OC0B is disabled.
2	DSX02	TC0 Trigger Source Select Control Enable bit 0. When the DSX02 bit is set to "1", the analog comparator 1 op amp channel 0 is used to turn off the output compare The trigger source for signal waveform OC0A / OC0B is enabled. When the DOC0A / DOC0B bit is "1" The rising edge of the interrupt flag register bit will automatically turn off the waveform of OC0A / OC0B. Out. When the DSX02 bit is set to "0", the analog comparator 1 op amp is set to 0 as the output compare The trigger source for signal waveform OC0A / OC0B is disabled.
1	DSX01	TC0 trigger source selection control enabled bit 1. When the DSX01 bit is set to "1", the analog comparator 0 op amp channel 1 is used to turn off the output compare The trigger source for signal waveform OC0A / OC0B is enabled. When the DOC0A / DOC0B bit is "1" The rising edge of the interrupt flag register bit will automatically turn off the waveform of OC0A / OC0B. Out. When setting the DSX01 bit to "0", the analog comparator 0 op amp channel 1 is used as the output compare The trigger source for signal waveform OC0A / OC0B is disabled.
0	DSX00	TC0 Trigger Source Select Control Enable bit 0. When setting the DSX00 bit to "1", the analog comparator 0 op amp channel 0 is set as the output compare The trigger source for signal waveform OC0A / OC0B is enabled. When the DOC0A / DOC0B bit is "1" The rising edge of the interrupt flag register bit will automatically turn off the waveform of OC0A / OC0B. Out. When setting the DSX00 bit to "0", the analog comparator 0 op amp channel 0 is used as the output compare

Page 92

The trigger source for signal waveform OC0A / OC0B is disabled.

The following table controls the selection of the trigger source for the waveform output.

Turn off the trigger source selection control of the OC0A / OC0B waveform output

DOC0x	DSX0n = 1	Trigger source	description
0	-	-	The DOC0x bit is "0" and the trigger source turns off the waveform output
			Can be forbidden
1	0	Analog Comparator 0 Channel 0	The rising edge of ACIF00 will turn off the OC0x waveform output
1	1	Analog Comparator 0 Channel 1	The rising edge of ACIF01 will turn off the OC0x waveform output
1	2	Analog Comparator 1 Channel 0	The rising edge of ACIF10 will turn off the OC0x waveform output
1	3	Analog Comparator 1 Channel 1	The rising edge of ACIF11 will turn off the OC0x waveform output
1	4	External interrupt 0	The rising edge of INTF0 will turn off the OC0x waveform output
1	5	Pin level change 0	The rising edge of PCIF0 will turn off the OC0x waveform output
1	6	TC2 overflow	The rising edge of TOV2 will turn off the OC0x waveform output
1	7	TC1 overflows	The rising edge of TOV1 will turn off the OC0x waveform output

note:

1) When DSX0n = 1 indicates that the nth bit of the DSX0 register is 1, each register bit can be set at the same time.

TC0 count value register **- TCNT0**

TCNT0 -***TC0*** count value register

Address: 0x46 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	TCNT07	TCNT06	TCNT05	TCNT04	TCNT03	TCNT02	TCNT01	TCNT00
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
		TC0 count value register.
		Through the TCNT0 register can be directly on the counter 8 for the count value to read and write access.
		The CPU writes to the TCNT0 register will prevent the compare match from the next timer clock cycle
		Occurs even if the timer has stopped. This allows the TCNT0 register to be initialized with the value of OCR0
7: 0	TCNT0	The value is consistent and does not trigger an interrupt.
		If the value written to TCNT0 equals or bypasses the OCR0 value, the compare match is lost, causing
		The correct waveform results.
		The timer stops counting when the clock source is not selected, but the CPU can still access TCNT0. CPU write count
		The device has a higher priority than zeroing or addition and subtraction.

TC0 output compare register **A-OCR0A**

OCR0A - ***TC0*** Output compare register ***A***

Address: 0x47 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	OCR0A7	OCR0A6	OCR0A5	OCR0A4	OCR0A3	OCR0A2	OCR0A1	OCR0A0

R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
		TC0 output compare register.
		OCR0A contains an 8-bit data that is continuously compared with the counter value TCNT0. Comparison
		The match can be used to generate an output compare interrupt, or to generate a waveform on the OC0A pin.
		When using PWM mode, the OCR0A register uses a double buffered register. While the normal working mode
7: 0	OCR0A	And the match clear mode, the double buffering function is disabled. Double buffering can be updated with OCR0A update
		Synchronize with the maximum or minimum moments of the count, thereby preventing the generation of asymmetric PWM pulses

Red, eliminating the interference pulse.
When using the double buffering function, the CPU accesses the OCR0A buffer register when the double buffering function is disabled

CPU access is OCR0A itself.

TC0 Output compare register **B- OCR0B**

OCR0B - **TC0** Output compare register **B**

Address: 0x48 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	OCR0B7	OCR0B6	OCR0B5	OCR0B4	OCR0B3	OCR0B2	OCR0B1	OCR0B0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	OCR0B	TC0 Output Compare B register. The OCR0B contains an 8-bit data that is continuously compared with the counter value TCNT0. The compare match can be used to generate an output compare interrupt, or to generate a waveform on the OC0B pin. When using PWM mode, the OCR0B register uses a double buffered register. While the ordinary work model Type and match clear mode, the double buffering function is disabled. Double buffering can be updated OCR0B The registers are synchronized with the count maximum or minimum moments to prevent asymmetry PWM pulse, eliminating the interference pulse. When using the double buffering function, the CPU accesses the OCR0B buffer register, which disables the double buffering function When the CPU accesses the OCR0B itself.

TC0 interrupt mask register - **TIMSK0**

TIMSK0 - **TC0** interrupt mask register

Address: 0x6E Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0
R / W	-	-	-	-	-	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description

7: 3		Keep it.
2	OCIE0B	TC0 Output Compare B Match Interrupt Enable bit. When the OCIE0B bit is "1" and the global interrupt is set, the TC0 output compare B matches the interrupt enable. when When a compare match occurs, that is, when the OCF0B bit in TIFR0 is set, an interrupt is generated. When the OCIE0B bit is "0", the TC0 output compare B match interrupt is disabled.
1	OCIE0A	TC0 Output Compare A Match Interrupt Enable bit. When the OCIE0A bit is "1" and the global interrupt is set, the TC0 output compares the A match interrupt enable. when When a compare match occurs, that is, when the OCF0A bit in TIFR0 is set, an interrupt is generated. When the OCIE0A bit is "0", the TC0 output compare A match interrupt is disabled.
0	TOIE0	TC0 overflow interrupt enable bit. When the TOIE0 bit is "1" and the global interrupt is set, the TC0 overflow interrupt is enabled. When TC0 overflows, When the TOV0 bit in TIFR is set, the interrupt is generated. When the TOIE0 bit is "0", the TC0 overflow interrupt is disabled.

TC0 Interrupt Flag Register - **TIFR0**

TIFR0 - **TC0** interrupt flag register

Address: 0x35 Default: 0x00

Bit	7	6	5	4	3	2	1	0

Name	OC0A	OC0B	-	-	-	OCF0B	OCF0A	TOV0
R / W	R / O	R / O	-	-	-	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	OC0A	Output compare waveform signal OC0A. Output compare waveform signal OC0A, software readable but not writable. The software can be enabled in the OC0A letter Before outputting to its corresponding IO pin, you can first read the value of the OC0A bit to get the output to be output Compare the polarity of the waveform signal and change it by configuring the COM0A bit and setting the FOC0A bit Its polarity, to avoid the OC0A signal output to its corresponding IO pin after the excess Interference pulse.
6	OC0B	Output compare waveform signal OC0B. Output compare waveform signal OC0B, software readable but not writable. The software can be enabled in OC0B messages Before the output to its corresponding IO pin, you can first read the value of the OC0B bit to get the output Compare the polarity of the waveform signal and change it by configuring the COM0B bit and setting the FOC0B bit Its polarity, to avoid the OC0B signal output to its corresponding IO pin after the excess Interference pulse.
5: 3		Keep it
2	OCF0B	TC0 Output Compare B match flag. When TCNT0 is equal to OCR0B, the comparison unit gives the match signal and sets the compare flag OCF0B. If the output compare B interrupt enable OCIE0B is "1" and the global interrupt flag is set Bit, an output compare B interrupt is generated. OCF0B will be cleared automatically when this interrupt service routine is executed Zero, or "1" on the OCF0B bit can also be cleared.
1	OCF0A	TC0 output compare A match flag. When TCNT0 is equal to OCR0A, the comparison unit gives the match signal and sets the compare flag

0	TOV0	OCF0A. If the output compare A interrupt is enabled, OCIE0A is set to "1" and the global interrupt flag is set Bit, an output compare A interrupt is generated. OCF0A will be cleared automatically when this interrupt service routine is executed Zero, or write "1" to the OCF0A bit. This bit can also be cleared. TC0 overflow flag. When the counter overflows, set the overflow flag TOV0. If the overflow interrupt is enabled, TOIE0 Is set to "1" and the global interrupt flag is set, an overflow interrupt is generated. Execute this interrupt service routine TOV0 will be cleared automatically, or "1" for TOV0 bits can also be cleared.

DTR0 - TC0 dead time control register

DTR0 - **TC0** dead time control register

Address: 0x4F Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	DTR07	DTR06	DTR05	DTR04	DTR03	DTR02	DTR01	DTR00
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	DTR0	TC0 dead time register. When the DTEN0 bit of the TCCR0B register is "1", the inserted dead time control is enabled. The dead time of the insertion is determined by DTR0, and the length of the time is the corresponding DTR0 count clock time.

TCKCSR - TC clock control and status register

TCKSCR - TC clock control and status register

Address: 0xEC Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
R / W	-	R / W	R	R	-	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	-	Keep it
6	F2XEN	RC 32M multiplier output enable control bit. When the F2XEN bit is set to "1", the multiplier output of the 32M RC oscillator is enabled and outputs 64M High speed clock. When the F2XEN bit is set to "1", the multiplier output of the 32M RC oscillator is disabled and can not be output 64M high speed clock.
5	TC2XF1	TC high speed clock mode flag bit 1. See Timer 1 Register Description.
4	TC2XF0	TC high speed clock mode flag 0. When reading the TC2XF0 bit as "1", it indicates that the timer counter 0 is operating in high-speed clock mode, "0" indicates that the timer counter 0 is operating in system clock mode.

Page 96

3	-	Keep it.
2	AFCKS	AC0 / 1 filter clock selection, please refer to OP / AC section for detailed definition
1	TC2XS1	TC high speed clock mode selection control bit 1. See Timer 1 Register Description.
0	TC2XS0	TC high speed clock mode selection control bit 0. When the TC2XS0 bit is set to "1", the timer counter 0 is selected to operate in the high-speed clock mode. When the TC2XS0 bit is set to "0", the timer counter 0 is selected to operate in the system clock mode.

16 -bit timer / counter *1*

True 16-bit design allows 16-bit PWM

- 2 independent output compare units
- Double buffered output compare register
- 1 input capture unit
- Input capture noise suppressor
- The counter is automatically cleared when the match matches and is automatically loaded
- Phase-corrected PWM without interference pulse
- Variable PWM cycle
- Frequency generator
- External event counter
- 4 independent interrupt sources
- PWM that supports dead time control
- Four optional trigger sources automatically turn off the PWM output

High-speed, high-resolution (500KHZ @ 7BIT) PWM in high-speed clock mode

Overview

TC1 structure diagram

TC1 is a universal 16-bit timer counter module that supports PWM output and can accurately generate waveforms. TC1 package
Includes 1 16-bit counter, waveform generation mode control unit, 2 independent output compare units and 1 input capture list
yuan. At the same time, TC1 can be shared with TC0 10-bit prescaler, you can also use the 10-bit prescaler. Prescaler
On the system clock clkio or high speed clock rcm2x (internal 32M RC oscillator output clock rc32m 2 times)
Divide to produce the count clock Clkt1. The waveform generation mode control unit controls the operation mode of the counter and the comparison output wave
The shape of the generation. Depending on the operating mode, the counter is cleared for each count clock Clkt1, plus one or minus
For. Clkt1 can be generated by an internal clock source or an external clock source. When the counter count value TCNT1 reaches the maximum value (etc.)
At maximum 0xFFFF or fixed value or output compare register OCR1A or input capture register ICR1, defined as TOP,
When the maximum value is MAX for the difference), the counter is cleared or decremented. When the counter counts the value TCNT1
When the minimum value (equal to 0x0000, defined as BOTTOM) is reached, the counter is incremented. When the counter counts
When the value TCNT1 reaches OCR1A or OCR1B, it is also called when the compare match is cleared or the output compare signal is set
OC1A or OC1B, to generate PWM waveforms. When the dead time is enabled, the set dead time (DTR1 is registered
The number of clocks corresponding to the device) will be inserted into the generated PWM waveform. When the input capture function is on, count
The ICR1 register will record the count value of the capture signal during the trigger period. The software is available
Clear the COM1A / COM1B bit to zero to turn off the OC1A / OC1B waveform output, or set the corresponding trigger source, when touch
When the event occurs, the hardware automatically clears the COM1A / COM1B bit to turn off the OC1A / OC1B waveform output.

The counting clock can be generated by an internal or external clock source. The clock source selection and frequency selection are set by the TCCR1B register
The CS1 bits of the device are controlled, as described in the TC0 and TC1 prescaler sections.

The length of the counter is 16 bits and supports bidirectional counting. The waveform generation mode is the operating mode of the counter located by TCCR1A
And the WGM1 bit of the TCCR1B register. According to the different operating modes, the counter for each count clock Clkt1
To achieve zero, plus one or minus one operation. When the count overflows, the count overflow flag TOV1 is located in the TIFR1 register
Bit will be set. TC1 count overflow interrupt can be generated when interrupt is enabled.

The output compare unit compares the count value TCNT1 with the values of the output compare registers OCR1A and OCR1B. When TCNT1
Equal to OCR1A or OCR1B is called a compare match, the output comparison flag OCF1A in the TIFR1 register or
The OCF1B bit is set. TC1 output compare match interrupt can be generated when interrupt is enabled.
Note that in the PWM operating mode, the OCR1A and OCR1B registers are double buffered registers. In normal mode
And CTC mode, the double buffering function is disabled. When the count reaches the maximum or minimum value, the value in the buffer register is synchronized
New to compare register OCR1A and OCR1B. See the description of the working mode section.

The waveform generator generates and controls the output mode control according to the waveform generation mode and uses the compare match and the count overflow to generate
Output the comparison waveform signals OC1A and OC1B. The specific way of generating is described in the working mode and register section description. To output
When comparing the waveform signals OC1A and OC1B to the corresponding pins, it is also necessary to set the data direction register of this pin to
Output.

Operating mode

Timing counter 1 has six different modes of operation, including normal mode (Normal), compare match clear (CTC)
Mode, fast pulse width modulation (FPWM) mode, phase corrected pulse width modulation (PCPWM) mode, phase frequency
Rate correction pulse width modulation (PFCPWM) mode, and input capture (ICP) mode. The waveform is generated by the mode control bit
WGM1 [3: 0] to choose. The six modes are described in detail below. Since there are two separate output comparison units, respectively
"A" and "B", with the lowercase "x" to represent the two output compare cell channels.

Normal mode

The normal mode is the simplest mode of operation for the timer counter. The waveform generation mode control bit, WGM1 [3: 0] = 0,
The maximum value of the TOP is MAX (0xFFFF). In this mode, the count mode increments by one for each count clock,
When the counter reaches the TOP overflow, it returns to BOTTOM to restart. The same value as the count value TCNT1 becomes zero
Set the timer counter overflow flag TOV1 in the count clock. The TOV1 flag in this mode is like the 17th bit, only
Will only be set to not be cleared. The overflow interrupt service routine automatically clears the TOV1 flag, which the software can use to improve it
Timing counter resolution. There is no special case in normal mode to consider, you can always write a new count value.

The waveform of the output compare signal OC1x can be obtained by setting the data direction register of the OC1x pin to be output. When COM1x = 1
, The OC1x signal is flipped when a compare match occurs. The frequency of the waveform in this case can be calculated using the following formula:

$f_{oc1xnormal} = f_{sys} / (2 * N * 65536)$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).

The output compare unit can be used to generate interrupts, but interrupts are not recommended in normal mode, which takes too much CPU
time.

CTC mode

When setting WGM1 [3: 0] = 4 or 12, timer counter 1 enters CTC mode. When WGM1 [3] = 0, the most count
The large value TOP is OCR1A, and when WGM1 [3] = 1, the maximum value TOP is ICR1. The following is the WGM1 [3: 0] = 4 for
Example To describe the CTC mode In this mode, the count mode is incremented by one for each count clock. When the counter value
The counter is cleared when TCNT1 is equal to TOP. This mode allows the user to easily control the frequency of the compare match output,
Also simplifies the operation of external event counting.

When the counter reaches TOP, the output compare match flag OCF1 is set and the corresponding interrupt enable is set
Off. The OCR1A register can be updated in the interrupt service routine. In this mode OCR1A does not use double buffering in
The counter should be careful when updating the maximum value to near minimum with no prescaler or very low prescaler operation. Such as
When the value of the write OCR1A is less than the current TCNT1 value, the counter will lose a compare match. In the next comparison
Before the match occurs, the counter has to count to MAX, and then count from BOTTOM to OCR1A. And general
The TOV1 flag is set in the count clock with the count value returned to 0x0.

The waveform of the output compare signal OC1x can be obtained by setting the data direction register of the OC1x pin to be output. The frequency of the waveform
The rate can be calculated using the following formula:

$f_{oc1xctc} = f_{sys} / (2 * N * (1 + OCR1A))$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).
As can be seen from the formula, when the set OCR1A is 0x0 and no prescaler, you can get the maximum frequency f_{sys} / 2 output
Waveform.

When WGM1 [3: 0] = 12 is similar to WGM1 [3: 0] = 4, it is only necessary to replace ICCR1 with OCR1A.

Fast PWM mode

When the WGM1 [3: 0] = 5, 6, 7, 14 or 15 is set, the timer counter 1 enters the fast PWM mode with the largest count
The values TOP are 0xFF, 0x1FF, 0x3FF, ICR1, or OCR1A, respectively, and can be used to generate high frequency PWM waveforms. fast
The PWM mode differs from other PWM modes in that it is a one-way operation. The counter is returned from the BOTTOM to the TOP
To BOTTOM to count again. When the count value TCNT1 reaches TOP or BOTTOM, the output compare signal OC1x will be
Set or clear, depending on the setting of the compare output mode COM1. See the register description for details. As a result of one-way operation,

The operating frequency of the fast PWM mode is twice the phase-corrected PWM mode using bidirectional operation. High frequency characteristics make fast
The speed PWM mode is ideal for power regulation, rectification, and DAC applications. High frequency signals can be reduced by external components (inductive power)
Capacity, etc.) size, thereby reducing system costs.

When the count reaches TOP, the timer counter overflow flag TOV1 will be set and the compare buffer value will be updated to
Compare values. If the interrupt is enabled, the OCR1A register can be updated in the interrupt service routine.

The waveform of the output compare signal OC1x can be obtained by setting the data direction register of the OC1x pin to be output. The frequency of the waveform
The rate can be calculated using the following formula:

$f_{oc1xfpwm} = f_{sys} / (N * (1 + TOP))$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).

When TCNT1 and OCR1x compare match, the waveform generator sets (clears) the OC1x signal. When TCNT1 is cleared
At zero, the waveform generator clears (sets) the OC1x signal to generate a PWM wave. Whereby the extreme value of OCR1x will be
Will produce a special PWM waveform. When OCR1x is set to 0x00, the output PWM is every (1 + TOP) count clock
There is a narrow spike. When OCR1x is set to TOP, the output waveform is a continuous high or low level.
If you use OCR1A as TOP and set COM1A = 1, the output compare signal OC1A will generate a duty cycle of 50% of the PWM
wave.

Phase correction PWM mode

When setting WGM0 [3: 0] = 1, 2, 3, 10 or 11, the timer counter 1 enters the phase correction PWM mode.
The maximum value of the number TOP is 0xFF, 0x1FF, 0x3FF, ICR1 or OCR1A. The counter is operated in two directions by BOTTOM
Increment to TOP, and then decrement it to BOTTOM, and repeat this operation. The count is changed when the count reaches TOP and BOTTOM
Number of directions, the count value in TOP or BOTTOM only stay a count clock. In the process of increment or decrement, count
When the value TCNT1 matches OCR1x, the output compare signal OC1x will be cleared or set, depending on the compare output mode
COM1 settings. Compared with one-way operation, the maximum frequency available for bi-directional operation is smaller, but its excellent symmetry is more
Combined with motor control.

In phase correction PWM mode, the TOV1 flag is set when the count reaches BOTTOM. When the count reaches TOP, compare
The value of the buffer is updated to the comparison value. If the interrupt is enabled, the compare buffer OCR1x can be updated in the interrupt service routine
Device.

The output compare signal OC1x waveform can be obtained by setting the data direction register of the OC1x pin to output. The frequency of the waveform can be
Use the following formula to calculate:

$f_{oc1xcpcpwm} = f_{sys} / (N * TOP * 2)$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).

During the up-counting process, the waveform generator clears (sets) the OC1x signal when TCNT1 matches OCR1x. in
In the process of decrementing the count, when the TCNT1 matches OCR1x, the waveform generator sets (clears) the OC1x signal. thus
The extreme value of OCR1x will produce a special PWM wave. When the OCR1x is set to TOP or BOTTOM, the OC1x signal is output
Will remain low or high. If OCR1A is used as TOP and COM1A = 1 is set, the compare signal OC1A is output
Will produce a duty cycle of 50% of the PWM wave.

In order to ensure that the output PWM wave symmetry on both sides of the BOTTOM, in the absence of a comparison match, there are two cases
Will flip the OC1x signal. The first case is when the value of OCR1x changes from TOP to other data. When OCR1x is
TOP, when the count value reaches TOP, the output of OC1x is the same as that of the previous descending order, that is, the OC1x
constant. The value of the new OCR1x (not TOP) is updated and the OC1x value remains until the ascending order
Count when the match occurs and flip. The OC1x signal is not centered at the minimum, so it needs to be in TCNT1
When the maximum value is reversed, the OC1x signal is flipped, which is the first case when the OC1x signal is flipped without a compare match. second
The case is that when TCNT1 starts counting from a value higher than OCR1x, it will lose a compare match, causing
The generation of asymmetric situations. Also need to flip OC1x signal to achieve the minimum on both sides of the symmetry.

Phase frequency correction PWM mode

When setting WGM0 [3: 0] = 8 or 9, the timer counter 1 enters the phase frequency correction PWM mode, counting the most
The large value TOP is ICR1 or OCR1A respectively. The counter is bi-directional, incremented by BOTTOM to TOP, and then passed
Reduced to BOTTOM, and repeat this operation. When the count reaches TOP and BOTTOM, the counting direction is changed and the count value is TOP
Or BOTTOM only stay on a count clock. In the process of increment or decrement, the count value TCNT1 matches OCR1x
, The output compare signal OC1x will be cleared or set, depending on the setting of the compare output mode COM1. With one-way operation
The maximum frequency available for bi-directional operation is smaller, but its excellent symmetry is more suitable for motor control.

Phase frequency correction In PWM mode, the TOV1 flag is set when the count reaches BOTTOM, and the compare buffer
Value is updated to the comparison value, the time for updating the comparison value is the phase frequency correction PWM mode and the phase correction PWM mode is the most
Big difference. If the interrupt is enabled, the compare buffer OCR1x register can be updated in the interrupt service routine. When the CPU changes
When the TOP value is ORC1A or ICR1, it is necessary to ensure that the new TOP value is not less than the TOP value that is already in use.
The comparison match will not happen again.

The output compare signal OC1x waveform can be obtained by setting the data direction register of the OC1x pin to output. The frequency of the waveform can be
Use the following formula to calculate:

$f_{oc1xcpfcpwm} = f_{sys} / (N * TOP * 2)$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).

During the up-counting process, the waveform generator clears (sets) the OC1x signal when TCNT1 matches OCR1x. in
In the process of decrementing the count, when the TCNT1 matches OCR1x, the waveform generator sets (clears) the OC1x signal. thus
The extreme value of OCR1x will produce a special PWM wave. When the OCR1x is set to TOP or BOTTOM, the OC1x signal is output
Will remain low or high. If OCR1A is used as TOP and COM1A = 1 is set, the compare signal OC1A is output
Will produce a duty cycle of 50% of the PWM wave.

Since the OCR1x register is updated at the BOTTOM time, the TOP value is on both sides in ascending and descending order.
Kind of symmetry waveforms where both the frequency and the phase are correct.

When using the fixed TOP value, it is best to use the ICR1 register as the TOP value, that is, set WGM1 [3: 0] = 8, then OCR1A

The register is only used to generate the PWM output. If you want to generate frequency changes in the PWM wave, you must change the TOP value,
OCR1A double buffering feature will be more suitable for this application.

Enter the capture mode

The input capture is used to capture an external event and give it a time stamp to indicate when the event occurred,
The counting mode is performed, but the waveform generation mode using the ICR1 value as the count TOP value is to be removed.

Trigger signals for external events are entered by pin ICP1 and can also be implemented by analog comparator units. When pin ICP1
The logic level changes, or the analog ACO level of the analog comparator changes, and this level change is lost
Captured by the capture unit, the input capture is triggered, then the 16-bit count value TCNT1 data is copied to the input capture
Register ICR1, and input capture flag ICF1 is set. If ICIE1 bit is "1", input capture flag will generate input capture
Catch interrupted.

The input capture is set by setting the analog compare control with the analog compare input control bit ACIC of the status register ACSR
Capture the source ICP1 or ACO. It should be noted that changing the trigger source may cause an input capture, so change the touch
A clear operation must be performed on ICF1 to avoid erroneous results.

The input capture signal is passed to an edge detector via an optional noise suppressor, and the control bit is selected according to the input capture
ICES1 configuration, to see whether the detected edge to meet the trigger conditions. The noise suppressor is a simple digital filter, right
The input signal is sampled four times, and the output is fed to the edge detector only when the four samples are equal. Noise suppression
The device is enabled or disabled by the ICNC1 bit of the TCCR1B register.

When the input capture function is used, the ICR1 register value should be read as early as possible when ICF1 is set, since the next time

The value of ICR1 will be updated after the capture event occurs. It is recommended to enable the input capture interrupt in any input capture mode , It is not recommended to change the count TOP value during operation.

The input capture time stamp can be used to calculate the frequency, duty cycle and other characteristics of the signal, as well as to trigger the event creation date Mind When measuring the duty cycle of an external signal, it is required to change the trigger edge after each capture. Therefore, after reading the ICR1 value, Quickly change the edge of the triggered signal.

Dead time control

When the DTEN1 bit is set to "1", the function of inserting the dead time is enabled and the output waveform of OC1A and OC1B will be B channel comparison output generated by the waveform based on the insertion of the set dead time, the length of the time for the DTR1 register The number of clocks corresponding to the time value. As shown in the following figure, the dead time insertion of OC1A and OC1B is the ratio of channel B Compared to the output waveform as a reference. When COM1A and COM1B are the same as "2" or "3", OC1A waveform polarity and OC1B Of the waveform polarity is the same, when COM1A and COM1B are "2" or "3", OC1A waveform and OC1B wave The opposite polarity.

Figure 3 TC1 dead time control in FPWM mode

Figure 4 TC1 dead time control in PCPWM mode

Page 104

When the DTEN1 bit is set to "0", the function of inserting dead time is disabled. The output waveform of OC1A and OC1B is the ratio
More than the output generated by the waveform.

High speed counting mode

In high-speed clock mode, a higher frequency clock is used as the counting clock source to generate higher speed and higher
Resolution of the PWM waveform. This high-frequency clock is performed by multiplying the output clock rc32m of the internal 32M RC oscillator by 2 times
To produce. Therefore, before entering the high-frequency mode, you must first enable the internal 32M RC oscillator multiplier function, that is set
TCKCSR register of the F2XEN bit and wait for a certain period of time until the multiplier clock signal output is stable. Then, it can be set
The TC2XS1 bit of TCKCSR enables the timer counter to enter high-speed clock mode.

In this mode, the system clock is asynchronous to the high-speed clock, and some registers (see the TC1 register list) are operating
In the high-speed clock domain, therefore, when configuring and reading such registers is also asynchronous, the operation should pay attention.

There is no special requirement for non-sequential read and write operations on registers in high-speed clock domains, and for continuous read and write operations
To wait for a system clock, follow these steps:
5) write register A;
6) wait for a system clock (NOP or operating system clock under the register);
7) Read or write to register A or B.
8) Wait for a system clock (NOP or register under operating system clock).

When a register is read in a high-speed clock domain, a register with a width of 8 bits can be read directly and 16 bits are read
Register value (OCR1A, OCR1B, ICR1, TCNT1), the value of the lower register is read first, waiting for a system clock
, Then read the value of the high register, and read the value of TCNT1, when the counter is still counting, TCNT1
Value will change with the high-speed clock, you can pause the counter (set CS1 to zero) and then read the value of TCNT1.

When reading OCR1A, OCR1B and ICR1, follow these steps:
1) Read OCR1AL / OCR1BL / ICR1L;
2) wait for a system clock (NOP);
3) Read OCR1AH / OCR1BH / ICR1H.

When reading TCNT1, follow these steps:
1) set CS1 to zero;
2) wait for a system clock (NOP);
3) read the value of TCNT1L;
4) wait for a system clock (NOP);
Read the value of TCNT1H.

Register definition

TC1 register list

register	address	Defaults	description
TCCR1A *	0x80	0x00	TC1 control register A
TCCR1B *	0x81	0x00	TC1 control register B
TCCR1C *	0x82	0x00	TC1 control register C
DSX1	0x83	0x00	TC1 Trigger Source Control Register
TCNT1L *	0x84	0x00	TC1 count value register low byte
TCNT1H *	0x85	0x00	TC1 count register high byte
ICR1L *	0x86	0x00	TC1 input capture register low byte
ICR1H *	0x87	0x00	TC1 input capture register high byte
OCR1AL *	0x88	0x00	TC1 Output compare register A low byte
OCR1AH *	0x89	0x00	TC1 Output compare register A high byte
OCR1BL *	0x8A	0x00	TC1 Output compare register B low byte
OCR1BH *	0x8B	0x00	TC1 Output compare register B high byte
DTR1 *	0x8C	0x00	TC1 dead time control register
TIMSK1	0x6F	0x00	The timer counter interrupts the mask register
TIFR1	0x36	0x00	Timer counter interrupt flag register
TCKCSR1	0xEC	0x00	TC1 clock control status register

【note】

The registers with "*" work in the system clock and the high-speed clock domain, and the registers without "*" work only in the system Under the clock domain.

TCCR1A -TC1 control register A

TCCR1A - TC1 control register A

Address: 0x80 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	COM1A1	COM1A0	COM1B1	COM1B0			WGM11	WGM10
R / W	R / W	R / W	R / W	R / W	-	-	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	COM1A1	Compare match output A mode control high. COM1A1 and COM1A0 form COM1A [1: 0] to control the output compare waveform OC1A. in case COM1A 1 or 2 are set, the output waveform occupies the OC1A pin, but the The pin's data direction register must be set high to output this waveform. In different working modes, COM1A on the output comparison waveform control is also different, see the specific comparison output mode control table description Described.
6	COM1A0	Compare match output A mode control low. COM1A1 and COM1A0 form COM1A [1: 0] to control the output compare waveform OC1A. in case

		COM1A 1 or 2 are set, the output waveform occupies the OC1A pin, but the The pin's data direction register must be set high to output this waveform. In different working modes,
		COM1A on the output comparison waveform control is also different, see the specific comparison output mode control table description Described.
		Compare match output B mode control high.
5	COM1B1	COM1B1 and COM1B0 form COM1B [1: 0] to control the output compare waveform OC1B. in case COM1B 1 or 2 are set, the output compare waveform occupies the OC1B pin, but the The pin's data direction register must be set high to output this waveform. In different working modes,
		COM1B on the output comparison waveform control is also different, see the specific comparison output mode control table Described.
		Compare match output B mode control low.
4	COM1B0	COM1B1 and COM1B0 form COM1B [1: 0] to control the output compare waveform OC1B. in case COM1B 1 or 2 are set, the output compare waveform occupies the OC1B pin, but the The pin's data direction register must be set high to output this waveform. In different working modes,
		COM1B on the output comparison waveform control is also different, see the specific comparison output mode control table Described.
3: 2	-	Keep it.
1	WGM11	Waveform generation mode control times low.
		WGM11 and WGM13, WGM12, WGM10 together constitute the waveform generation mode control WGM1 [3: 0], control counter count mode and waveform generation mode, see the waveform generation model Formula Description.
0	WGM10	The waveform generation mode controls the least significant bit.
		WGM10 and WGM13, WGM12, WGM11 together constitute the waveform generation mode control WGM1 [3: 0], control counter count mode and waveform generation mode, see the waveform generation model Formula Description.

The following table shows the non-PWM mode (ie, normal mode and CTC mode), compare the output mode to the output compare waveform control.

COM1x [1: 0]	description
0	OC1x disconnect, general IO port operation
1	Compare the OC1x signal when matching
2	The OC1x signal is cleared when compare match
3	Set OC1x signal when compare match

The following table shows the control of the output compare waveform for the compare output mode in fast PWM mode.

COM1x [1: 0]	description
0	OC1x disconnect, general IO port operation
1	When WGM1 is 15: The OC1A signal is flipped when compare match, OC1B is off When WGM1 is other value: OC1x is disconnected, general purpose IO port operation
2	The OC1x signal is cleared when the compare match is set and the OC1x signal is set when the maximum match is made
3	The OC1x signal is set when the compare match is cleared and the OC1x signal is cleared when the maximum match is made

The following table shows the control of the output compare waveform for the compare output mode in phase correction mode.

COM1x [1: 0]	description
0	OC1x disconnect, general IO port operation

1	When WGM1 is 9 or 11: OC1A signal is flipped when compare match, OC1B is off When WGM1 is other value: OC1x is disconnected, general purpose IO port operation
2	Ascending Count Compare Match Clear OC1x Signal, Compare Descending Compare Count Set OC1x Signal

3 Ascending Count Compare Match Set OC1x Signal, Compare Descending Compare Descending Cleared OC1x Signal

TCCR1B -TC1 control register **B**

TCCR1B - TC1 Control Register B

Address: 0x81 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
R / W	R / W	R / W	-	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	ICNC1	The input capture noise suppressor enables the control bit. When the ICNC1 bit is set to "1", the input capture noise suppressor is enabled, and the external pin ICP1 Of the input is filtered, the input signal is valid when four consecutive samples are equal, which makes the input The capture was delayed by four clock cycles. When the ICNC1 bit is set to "0", the input capture noise suppressor is disabled, and the external pin ICP1 The input is directly valid.
6	ICES1	The input capture trigger edge selects the control bit. When the ICES1 bit is set to "1", the rising edge of the selection level triggers the input capture; when ICES1 is set When bit is "0", the falling edge of the selection level triggers the input capture. When an event is captured, the counter value is copied to the ICR1 register and the input is set Capture logo ICF1. If the interrupt is enabled, an input capture interrupt is generated.
5	-	Keep it.
4	WGM13	Waveform Generation Mode Control High. WGM13 and WGM12, WGM11, WGM10 together constitute the waveform generation mode control WGM1 [3: 0], control counter count mode and waveform generation mode, see the waveform generation mode Table description.
3	WGM12	Waveform generation mode control times high. WGM12 and WGM13, WGM11, WGM10 together constitute the waveform generation mode control WGM1 [3: 0], control counter count mode and waveform generation mode, see the waveform generation mode Table description.
2	CS12	Clock select control high. Used to select the clock source for timer counter 1.
1	CS11	Clock selection control bit. Used to select the clock source for timer counter 1.
0	CS10	Clock select control low. Used to select the clock source for timer counter 1.

CS1 [2: 0]	description
0	No clock source, stop counting
1	clk $_{sys}$
2	clk $_{sys}$ / 8, from prescaler

3	clk $_{sys}$ / 64, from prescaler
4	clk $_{sys}$ / 256, from prescaler
5	clk $_{sys}$ / 1024, from prescaler
6	External clock T1 pin, falling edge trigger
7	External clock T1 pin, rising edge triggered

The following table shows the waveform generation mode control.

WGM1 [3: 0] working mode		**TOP** value	Update the time of **TOCR0** when the **OCR0 is** updated	
0	Normal	0xFFFF	immediately	MAX
1	8 bit PCPWM	0x00FF	TOP	BOTTOM
2	9 bit PCPWM	0x01FF	TOP	BOTTOM

3	10 bit PCPWM	0x03FF	TOP	BOTTOM
4	CTC	OCR1A	immediately	MAX
5	8-bit FPWM	0x00FF	BOTTOM	TOP
6	9 bits FPWM	0x01FF	BOTTOM	TOP
7	10 bits FPWM	0x03FF	BOTTOM	TOP
8	PFCPWM	ICR1	BOTTOM	BOTTOM
9	PFCPWM	OCR1A	BOTTOM	BOTTOM
10	PCPWM	ICR1	TOP	BOTTOM
11	PCPWM	OCR1A	TOP	BOTTOM
12	CTC	ICR1	immediately	MAX
13	Keep it	-	-	-
14	FPWM	ICR1	TOP	TOP
15	FPWM	OCR1A	TOP	TOP

TCCR1C -TC1 control register **C**

TCCR1C - TC1 control register

Address: 0x82 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	FOC1A	FOC1B	DOC1B	DOC1A	DTEN1		-	-	-
R / W	W	W	R / W	R / W	R / W	-	-	-
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
		Force Output Compare A.
		When operating in non-PWM mode, the "1" mode can be written to the forced output compare bit FOC1A
		To produce a comparison match. Forcing the compare match does not set the OCF1A flag, nor is it overloaded or cleared
7	FOC1A	Timer, but the output pin OC1A will be updated according to the COM1A settings accordingly
		The same happens as a match.
		When working in PWM mode, write the TCCR1A register to clear it.
		The return value of the read FOC1A is always zero.

6	FOC1B	Force the output comparison B.
		When operating in non-PWM mode, it is possible to write "1" to the forced output compare bit FOC1B
		To produce a comparison match. Forcing the match does not set the OCF1B flag, nor is it overloaded or cleared
		Timer, but the output pin OC1B will be updated according to the COM1B settings accordingly
		The same happens as a match. When working in PWM mode, write the TCCR1A register to its
		Cleared. The return value of the read FOC1B is always zero.
5	DOC1B	TC1 OFF Output Compare Enable Control High.
		When the DOC1B bit is set to "1", the trigger source off output compare signal OC1B is enabled. When made
		When the event is triggered, the hardware automatically turns off the OC1B's waveform output.
		When the DOC1B bit is set to "0", the trigger source off output compare signal OC1B is disabled. When made
		When the event is triggered, the OC1B waveform output is not turned off.
4	DOC1A	TC1 OFF Output Compare Enable Control Low.
		When the DOC1A bit is set to "1", the trigger source off output compare signal OC1A is enabled. When made
		When the event is triggered, the hardware automatically turns off the OC1A's waveform output.
		When the DOC1A bit is set to "0", the trigger source off output compare signal OC1A is disabled. When made
		When the event is triggered, the OC1A's waveform output is not turned off.
3	DTEN1	TC1 dead time enable control bit.
		When the DTEN1 bit is set to "1", the dead time is enabled. OC1A and OC1B are in the B channel
		Compare the output generated by the waveform based on the insertion dead time, the inserted dead time interval by the DTR1
		The count time corresponding to the register is determined. OC1A output waveform polarity by COM1A and COM1B

		The relationship between the decision, see OC1A insert dead time after the waveform polarity table shown.
		When the DTEN1 bit is set to "0", the dead time insertion is disabled and the waveforms of OC1A and OC1B are
		Since the comparison produces the resulting waveform.
2: 0	-	Keep it

The following table shows the polarity control of the OC1A signal output waveform when the dead time is enabled.

Polarity Control of OC1A Signal Output Waveform in Dead Time Enabled Mode

DTEN1	COM1A [1: 0]	COM1B [1: 0]	description
0	-	-	The OC1A signal polarity is controlled by the OC1A compare output mode
1	0	-	OC1A disconnect, general IO port operation
1	1	-	Keep it
1	2	2	The OC1A signal is the same polarity as the OC1B signal
		3	The OC1A signal is opposite to the polarity of the OC1B signal
1	3	2	The OC1A signal is opposite to the polarity of the OC1B signal
		3	The OC1A signal is the same polarity as the OC1B signal

note:

1) The polarity of the OC1B signal output waveform is controlled by the OC1B compare output mode, which is the same as the unimplemented dead time mode.

DSX1 -TC1 trigger source control register

DSX1 - TC trigger source control register

Address: 0x83 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	DSX17	DSX16	DSX15	DSX14	DSX13	DSX12	DSX11	DSX10

R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
		TC1 trigger source selection control enabled bit 7.
		When setting the DSX17 bit to "1", TC0 overflows as the output waveform
		OC1A / OC1B trigger source is enabled. When the DOC1A / DOC1B bit is "1", the selected touch
7	DSX17	The rising edge of the interrupt flag register bit will automatically turn off the OC1A / OC1B waveform input
		Out.
		When setting the DSX17 bit to "0", TC0 overflows as the output waveform
		The trigger source for OC1A / OC1B is disabled.
		TC1 Trigger Source Select Control Enable bit 6.
		When the DSX16 bit is set to "1", TC2 overflows as the output signal
		OC1A / OC1B trigger source is enabled. When the DOC1A / DOC1B bit is "1", the selected touch
6	DSX16	The rising edge of the interrupt flag register bit will automatically turn off the OC1A / OC1B waveform input
		Out.
		When the DSX16 bit is set to "0", TC2 overflows as the output waveform
		The trigger source for OC1A / OC1B is disabled.
		TC1 Trigger Source Select Control Enable Bit 5.
		When the DSX15 bit is set to "1", the pin level changes by 1 as the output signal
		The trigger source for waveform OC1A / OC1B is enabled. When the DOC1A / DOC1B bit is "1"
5	DSX15	The rising edge of the interrupt flag register bit of the trigger source automatically turns off the OC1A / OC1B wave
		Shaped output.
		When the DSX15 bit is set to "0", the pin level changes by 1 as the output signal
		The trigger source for waveform OC1A / OC1B is disabled.
		TC1 trigger source selection control enabled bit 4.
		When the DSX14 bit is set to "1", the external interrupt 1 is used to turn off the output compare signal waveform
		OC1A / OC1B trigger source is enabled. When the DOC1A / DOC1B bit is "1", the selected touch

4	DSX14	The rising edge of the interrupt flag register bit will automatically turn off the OC1A / OC1B waveform input Out. When the DSX14 bit is set to "0", the external interrupt 1 is used to turn off the output compare signal waveform The trigger source for OC1A / OC1B is disabled.
3	DSX13	TC1 Trigger Source Select Control Enable Bit 3. When the DSX13 bit is set to "1", the analog comparator 1 transports channel 1 as the output The trigger signal for the comparison signal waveform OC1A / OC1B is enabled. When the DOC1A / DOC1B bit is "1" , The rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off OC1A / OC1B The waveform output. When the DSX13 bit is set to "0", the analog comparator 1 is shipped with channel 1 as the output The trigger signal for the comparison signal waveform OC1A / OC1B is disabled.
2	DSX12	TC1 Trigger Source Select Control Enable Bit 0. When the DSX12 bit is set to "1", the analog comparator 1 operates as a function to turn off the output The trigger signal for the comparison signal waveform OC1A / OC1B is enabled. When the DOC1A / DOC1B bit is "1" , The rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off OC1A / OC1B The waveform output.

1	DSX11	When the DSX12 bit is set to "0", the analog comparator 1 transports channel 0 as the output The trigger signal for the comparison signal waveform OC1A / OC1B is disabled. TC1 Trigger Source Select Control Enable bit 1. When the DSX11 bit is set to "1", the analog comparator 0 op amp channel 1 is used as the output The trigger signal for the comparison signal waveform OC1A / OC1B is enabled. When the DOC1A / DOC1B bit is "1" , The rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off OC1A / OC1B The waveform output. When the DSX11 bit is set to "0", the analog comparator 0 op amp channel 1 is set as off output The trigger signal for the comparison signal waveform OC1A / OC1B is disabled.
0	DSX10	TC1 Trigger Source Select Control Enable Bit 0. When the DSX10 bit is set to "1", the analog comparator 0 op amp channel 0 is set as off output The trigger signal for the comparison signal waveform OC1A / OC1B is enabled. When the DOC1A / DOC1B bit is "1" , The rising edge of the interrupt flag register bit of the selected trigger source will automatically turn off OC1A / OC1B The waveform output. When the DSX10 bit is set to "0", the analog comparator 0 op amp channel 0 is used to turn off the output The trigger signal for the comparison signal waveform OC1A / OC1B is disabled.

The following table controls the selection of the trigger source for the waveform output.

Table 1 Turns off the trigger source selection control for the OC1A / OC1B waveform output

DOC1x	DSX1n = 1	Trigger source	description
0	-	-	The DOC1x bit is "0" and the trigger source turns off the waveform output Can be forbidden
1	0	Analog Comparator 0 Channel 0	The rising edge of ACIF00 will turn off the OC1x waveform output
1	1	Analog Comparator 0 Channel 1	The rising edge of ACIF01 will turn off the OC1x waveform output
1	2	Analog Comparator 1 Channel 0	The rising edge of ACIF10 will turn off the OC1x waveform output
1	3	Analog Comparator 1 Channel 1	The rising edge of ACIF11 will turn off the OC1x waveform output
1	4	External interrupt 1	The rising edge of INTF1 will turn off the OC1x waveform output
1	5	Pin level change	The rising edge of PCIF1 will turn off the OC1x waveform output
1	6	TC2 overflow	The rising edge of TOV2 will turn off the OC1x waveform output
1	7	TC0 overflows	The rising edge of TOV0 will turn off the OC1x waveform output

note:

2) When DSX1n = 1 indicates that the nth bit of the DSX1 register is 1, each register bit can be set at the same time.

TCNT1L -TC1 Count Value Register Low byte

TCNT1L - TC1 count register low byte

Address: 0x84 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	TCNT1L 7	TCNT1L 6	TCNT1L 5	TCNT1L 4	TCNT1L 3	TCNT1L 2	TCNT1L 1	TCNT1L 0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	TCNT1 [7: 0]	TC1 The low byte of the count value. TCNT1H and TCNT1L together to form TCNT1, through TCNT1 register can be directly Read and write access to the 16-bit count of the counter. Reading and writing 16-bit registers requires two operations. When writing 16-bit TCNT1, write TCNT1H first. When reading 16-bit TCNT1, it should be read first TCNT1L. The CPU writes to the TCNT1 register will prevent the compare horses from the next timer clock cycle With the occurrence, even if the timer has stopped. This allows the value of the TCNT1 register to be initialized Consistent with the value of OCR1x without triggering an interrupt. If the value of TCNT1 is written to or equal to the value of OCR1x, the comparison match is lost. Into an incorrect waveform. The timer stops counting when no clock source is selected, but the CPU can still access TCNT1. CPU write The counter has a higher priority than zeroing or addition and subtraction.

TCNT1H -TC1 count register high byte

TCNT1H - TC1 count register high byte

Address: 0x85 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	TCNT1H 7	TCNT1H 6	TCNT1H 5	TCNT1H 4	TCNT1H 3	TCNT1H 2	TCNT1H 1	TCNT1H 0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	TCNT1 [15: 8]	TC1 The high byte of the count value. TCNT1H and TCNT1L together to form TCNT1, through the TCNT1 register can be straight Read and write access to the 16-bit count of the counter. Reading and writing 16-bit registers takes twice operating. When writing 16-bit TCNT1, write TCNT1H first. When reading 16-bit TCNT1, first Read TCNT1L. The CPU writes to the TCNT1 register will prevent the compare horses from the next timer clock cycle With the occurrence, even if the timer has stopped. This allows the value of the TCNT1 register to be initialized Consistent with the value of OCR1x without triggering an interrupt. If the value written to TCNT1 equals or bypasses the OCR1x value, the compare match is lost, Resulting in incorrect waveform results. The timer stops counting when no clock source is selected, but the CPU can still access TCNT1. CPU write The counter has a higher priority than zeroing or addition and subtraction.

ICR1L -TC1 input capture register low byte

ICR1L - TC1 input capture register low byte

Address: 0x86 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	ICR1L7	ICR1L6	ICR1L5	ICR1L4	ICR1L3	ICR1L2	ICR1L1	ICR1L0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	ICR1 [7: 0]	TC1 Enter the low byte of the capture value. ICR1H and ICR1L combine to form 16-position ICR1. Reading and writing 16-bit registers requires two Operation. When writing 16-bit ICR1, write ICR1H first. When reading 16-bit ICR1, read it first ICR1L. When the input capture is triggered, the count value TCNT1 is updated to the ICR1 register In the device. The ICR1 register can also be used to define the TOP value of the count.

ICR1H -TC1 input capture register high byte

ICR1H - TC1 input capture register high byte

Address: 0x87 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	ICR1H7	ICR1H6	ICR1H5	ICR1H4	ICR1H3	ICR1H2	ICR1H1	ICR1H0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	ICR1 [15: 8]	TC1 Enter the high byte of the capture value. ICR1H and ICR1L combine to form 16-position ICR1. Reading and writing 16-bit registers requires two Operation. When writing 16-bit ICR1, write ICR1H first. When reading 16-bit ICR1, read it first ICR1L. When the input capture is triggered, the count value TCNT1 is updated to the ICR1 register In the device. The ICR1 register can also be used to define the TOP value of the count.

OCR1AL -TC1 Output compare register **A** low byte

OCR1AL - TC1 Output compare register A low byte

Address: 0x88 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	OCR1AL 7	OCR1AL 6	OCR1AL 5	OCR1AL 4	OCR1AL 3	OCR1AL 2	OCR1AL 1	OCR1AL 0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	OCR1A [7: 0]	Output the lower byte of compare register A. OCR1AL and OCR1AH combine to form 16-position OCR1A. Read and write 16 bits The register requires two operations. When writing 16-bit OCR1A, write OCR1AH first.

When reading 16-bit OCR1A, read OCR1AL first.

OCR1A is compared with the counter value TCNT1 without interruption. Compare matches can be used
To produce an output compare interrupt, or to generate a waveform on the OC1A pin.

When using PWM mode, the OCR1A register uses a double buffered register. And ordinary
In the mode of operation and match clear mode, the double buffering function is disabled. Double buffering
To synchronize the update OCR1A register with the count maximum or minimum time,
Thereby preventing the generation of asymmetric PWM pulses, eliminating the interference pulses.
When using the double buffering function, the CPU accesses the OCR1A buffer register, which is disabled
When the CPU function is OCR1A itself.

OCR1AH -TC1 Output compare register **A** high byte

OCR1AH - TC1 Output compare register A high byte

Address: 0x89 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	OCR1A H7	OCR1A H6	OCR1A H5	OCR1A H4	OCR1A H3	OCR1A H2	OCR1A H1	OCR1A H0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	OCR1A [15: 8]	Outputs the high byte of compare register A. OCR1AL and OCR1AH combine to form 16-position OCR1A. Read and write 16 bits The register requires two operations. When writing 16-bit OCR1A, write OCR1AH first. When reading 16-bit OCR1A, read OCR1AL first. OCR1A is compared with the counter value TCNT1 without interruption. Compare matches can be used To produce an output compare interrupt, or to generate a waveform on the OC1A pin. When using PWM mode, the OCR1A register uses a double buffered register. And ordinary In the mode of operation and match clear mode, the double buffering function is disabled. Double buffering To synchronize the update OCR1A register with the count maximum or minimum time, Thereby preventing the generation of asymmetric PWM pulses, eliminating the interference pulses. When using the double buffering function, the CPU accesses the OCR1A buffer register, which is disabled When the CPU function is OCR1A itself.

OCR1BL -TC1 Output compare register **B** low byte

OCR1BL - TC1 Output Compare Register B Low Byte

Address: 0x8A Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	OCR1BL 7	OCR1BL 6	OCR1BL 5	OCR1BL 4	OCR1BL 3	OCR1BL 2	OCR1BL 1	OCR1BL 0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W

Initial	0	0	0	0	0	0	0	0

Bit	Name	description
		Output the lower byte of compare register B.
		OCR1BL and OCR1BH combine to form 16-bit OCR1B. Read and write 16-bit registers
		Need two operations. When writing 16-bit OCR1B, write OCR1BH first. Read 16-bit OCR1B
		, You should first read OCR1BL.
		OCR1B is compared with the counter value TCNT1 without interruption. Compare matches can be used to generate

7: 0 OCR1B [7: 0]	Output compare interrupt, or used to generate waveforms on OC1B pins. When using PWM mode, the OCR1B register uses a double buffered register. While the ordinary work model Type and match clear mode, the double buffering function is disabled. Double buffering can be updated The OCR1B register is synchronized with the count maximum or minimum time to prevent generation of Symmetric PWM pulses eliminate the interference pulses. When using the double buffering function, the CPU accesses the OCR1B buffer register, which prohibits double buffering When the CPU access is OCR1B itself.

OCR1BH -TC1 Output compare register **B** high byte

OCR1BH - TC1 Output Compare Register B High Byte

Address: 0x8B Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	OCR1BH7	OCR1BH6	OCR1BH5	OCR1BH4	OCR1BH3	OCR1BH2	OCR1BH1	OCR1BH0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	OCR1B [15: 8]	Outputs the high byte of compare register B. OCR1BL and OCR1BH combine to form 16-bit OCR1B. Read and write 16-bit register The device needs two operations. When writing 16-bit OCR1B, write OCR1BH first. Read 16 bits OCR1B should read OCR1BL first. OCR1B is compared with the counter value TCNT1 without interruption. Compare matches can be used Generates an output compare interrupt, or is used to generate a waveform on the OC1B pin. When using PWM mode, the OCR1B register uses a double buffered register. While the general work In mode and match clear mode, the double buffering function is disabled. Double buffering can be more The new OCR1B register is synchronized with the counting of the maximum or minimum moments to prevent production The asymmetric PWM pulse eliminates the interference pulse. When using the double buffering function, the CPU accesses the OCR1B buffer register, which prohibits double buffering Function when the CPU access is OCR1B itself.

TIMSK1 - TC1 interrupt mask register

TIMSK1 - **TC1** interrupt mask register

Page 116

Address: 0x6F Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	TICIE1	-	-	OCIE1A	OCIE1B	TOIE1
R / W	-	-	R / W	-	-	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 6	-	Keep it.
5	TICIE1	TC1 input capture interrupt enable bit. When the ICIE1 bit is "1" and the global interrupt is set, the TC1 input capture interrupt is enabled. when When the input capture trigger is enabled, the ICF1 flag of TIFR1 is set and an interrupt occurs. When the ICIE1 bit is "0", the TC1 input capture interrupt is disabled.
4: 3	-	Keep it.
		TC1 Output Compare B Match Interrupt Enable bit. When the OCIE1B bit is "1" and the global interrupt is set, the TC1 output compares the B match interrupt enable.

2	OCIE1B	When a compare match occurs, that is, when the OCF1B bit in TIFR is set, an interrupt is generated. When the OCIE1B bit is "0", the TC1 output compare B match interrupt is disabled.
1	OCIE1A	TC1 Output Compare A Match Interrupt Enable bit. When the OCIE1A bit is "1" and the global interrupt is set, the TC1 output compares the A match interrupt enable. When a compare match occurs, that is, when the OCF1A bit in TIFR is set, an interrupt is generated. When the OCIE1A bit is "0", the TC1 output compare A match interrupt is disabled.
0	TOIE1	TC1 overflow interrupt enable bit. When the TOIE1 bit is "1" and the global interrupt is set, the TC1 overflow interrupt is enabled. When TC1 overflows When the TOV1 bit in TIFR is set, the interrupt is generated. When the TOIE1 bit is "0", the TC1 overflow interrupt is disabled.

TIFR1 - TC1 interrupt flag register

TIFR1 - **TC1** interrupt flag register

Address: 0x36 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1
R / W	-	-	R / W	-	-	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 6	-	Keep it.
5	ICF1	Enter the capture flag. The ICF1 flag is set when an input capture event occurs. When ICR1 is used as the TOP value of the count, And the count value reaches the TOP value, the ICF1 flag is set. If ICIE1 is "1" and the global interrupt flag is set Set, an input capture interrupt is generated. ICF1 is automatically cleared when this interrupt service routine is executed, Or write "1" to ICF1 bit to clear this bit.

4: 3	-	Keep it.
2	OCF1B	Output Compare B match flag. When TCNT1 is equal to OCR1B, the compare unit gives the match signal and sets the compare flag OCF1B. If the output compare interrupt enable OCIE1B is "1" and the global interrupt flag is set, Out of comparison. OCF1B will be cleared automatically or written to the OCF1B bit when this interrupt service routine is executed "1" can also be cleared to this bit.
1	OCF1A	Output Compare A match flag. When TCNT1 is equal to OCR1A, the compare unit gives the match signal and sets the compare flag OCF1A. If the output compare interrupt enable OCIE1A is "1" and the global interrupt flag is set, Out of comparison. OCF1A will be cleared automatically or written to OCF1A when this interrupt service routine is executed "1" can also be cleared to this bit.
0	TOV1	Overflow flag. When the counter overflows, set the overflow flag TOV1. If the overflow interrupt is enabled, TOIE1 is enabled "1" and the global interrupt flag is set, an overflow interrupt is generated. When this interrupt service routine is executed, TOV1 This bit can also be cleared by clearing it automatically or by writing "1" to the TOV1 bit.

DTR1 -TC1 dead time register

DTR1 - TC1 dead time register

Address: 0x8C Default: 0x00

Bit	7	6	5	4	3	2	1	0

Name	DTR17	DTR16	DTR15	DTR14	DTR13	DTR12	DTR1	DTR10
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	DTR1	TC1 dead time register. When the DTEN1 bit of the TCCR1B register is "1", the inserted dead time control is enabled. The dead time of the insertion is determined by DTR1, and the length of the time is the corresponding DTR1 count clock time.

TCKCSR -TC clock control status register

TCKCSR - TC clock control status register

Address: 0xEC Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
R / W	-	R / W	R / O	R / O	-	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	-	Keep it
6	F2XEN	RC 32M multiplier output enable control bit When the F2XEN bit is set to "1", the multiplier output of the 32M RC oscillator is enabled and outputs 64M High speed clock When the F2XEN bit is set to "1", the multiplier output of the 32M RC oscillator is disabled and can not be output 64M high speed clock
5	TC2XF1	TC high speed clock mode flag bit 1 When the TC2XF1 bit is read as "1", it indicates that the timer counter 1 is operating in the high-speed clock mode, "0", it indicates that the timer counter 1 is operating in the system clock mode
4	TC2XF0	TC High Speed Clock Mode Flag Bit 0, Reference Timer Counter 0 Register Description
3	-	Keep it
2	AFCKS	OP / AC filter clock selection, please refer to OP / AC section for detailed definition
1	TC2XS1	TC high speed clock mode selection control bit 1 When the TC2XS1 bit is set to "1", the timer counter 1 is selected to operate in high-speed clock mode When the TC2XS1 bit is set to "0", the timer counter 1 is selected to operate in the system clock mode
0	TC2XS0	TC High Speed Clock Mode Select Control Bit 0, Reference Timer Counter 0 Register Description

Timer / Counter *0/1* prescaler

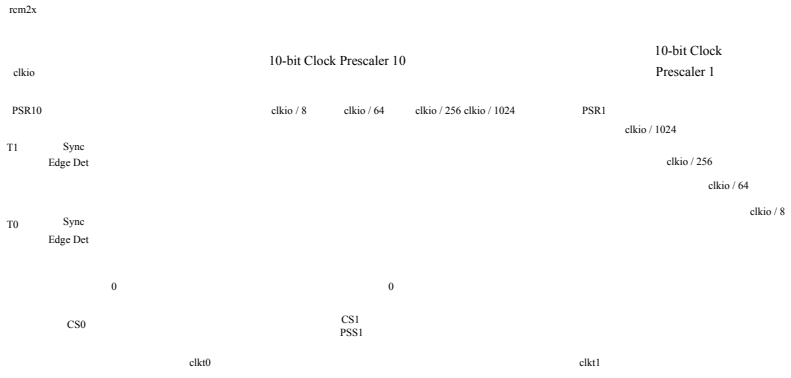
- 2 10-bit prescaler
- TC0 and TC1 multiplexed prescaler CPS10 in multiplexed mode
- In the single mode TC0 exclusive prescaler CPS10, TC1 exclusive prescaler CPS1
- Support software reset

Overview

In multiplexed mode (PSS1 = 0), TC0 and TC1 share a 10-bit prescaler CPS10, but they have different Frequency setting.

Single mode (PSS1 = 1), TC0 independent use of prescaler CPS10, TC1 independent use of prescaler CPS1, it They have different frequency settings.

The following description is used for TC0 and TC1, where n represents 0 or 1.



TC0 / TC1 Prescaler structure

Internal clock source

When setting CSn [2: 0] = 1, the timer counter can be set directly by the system clock clkio or the high speed clock rcm2x (internal 32M RC oscillator output clock 2 times the frequency) drive. Prescaler can output 4 different clock frequencies, respectively clkio / 8, clkio / 64, clkio / 256 and clkio / 1024.

The divider is reset

Multiplexing mode

When the PSS1 bit is set to "0", TC0 and TC1 share a prescaler CPS10.

The prescaler is run independently and its operation is independent of TC's clock selection logic, and it has TC0 and TC1 shared.

The status of the prescaler has an effect on the application of the frequency divider due to the influence of clock selection control. When the timer

When the output of the prescaler is enabled and the count clock source (6> CSn [2: 0]> 1) is selected, the effect is generated. From

The timer enable to the first count may take 1 to N + 1 system clocks, where N is the prescaler factor (8,

64,256 or 1024).

It is possible to synchronize the timer and program operation by resetting the prescaler. But it must be noted that another timer is

If you are using this prescaler, resetting the prescaler will affect all timers connected to it.

Single mode

When the PSS1 bit is set to "1", TC0 independently uses the prescaler CPS10, the prescaler is reset by the PSR10 bit

To control. TC1 independent use of prescaler CPS1, prescaler reset by the PSR1 bit to control. The respective reset

Acting alone, does not affect other prescalers.

External clock source

The external clock source provided by the T0 / T1 pin can be used as the count clock source. The signal on the T0 / T1 pin is synchronized

And the edge detector is used as the clock source for the counter. Each rising edge (CSn [2: 0] = 7) or falling edge (CSn [2: 0] = 6)

Will produce a count pulse. The external clock source is not fed into the prescaler.

Due to the presence of pin synchronization and edge detection circuitry, the change in T0 / T1 level requires a delay of 2.5 to 3.5 lines

The clock can update the counter.

Disable or enable the clock input must be held at T0 / T1 at least one system clock cycle before it can be performed.

The possibility of generating an error counting clock pulse.

To ensure proper sampling, the external clock pulse width must be greater than one system clock cycle, with a duty cycle of 50%

The external clock frequency must be less than half the system clock frequency. Due to the error of the oscillator itself brings the system clock frequency

And the difference in duty cycle, it is recommended that the maximum frequency of the external clock should not be greater than f_{sys} / 2.5.

Register definition

GTCCR - General Timing Counter Control Register

GTCCR - General Timing Counter Control Register

Address: 0x43 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	TSM	-	-	-	-	-	PSRASY	PSRSYNC
R / W	R / W	R / W	R / W	R / W	R / W	R / W	W	W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
		Timer counter synchronization mode control bits.
7	TSM	When setting the TSM bit to "1", it is the timer counter synchronization mode. In synchronous mode, write to PSRASY
		Bit and PSRSYNC bit values are held so that the corresponding prescaler is reset. This ensures that the corresponding

		The timer counter is aborted and configured to the same value.
		When the TSM bit is set to "0", the values of the PSRASY and PSRSYNC bits are cleared by hardware and
		When the counter starts working at the same time.
6: 2	-	Keep it.
1	PSRASY	See Timer TC2 Register Description.
0	PSRSYNC	Prescaler CPS10 reset control bit.
		When the PSRSYNC bit is set to "1", the prescaler CPS10 will be reset. When the TSM bit is not set
		, The hardware clears the PSRSYNC bit after reset.
		When the PSRSYNC bit is set to "0", the setting is invalid.
		In multiplexed mode, the TC0 / TC1 shared prescaler, the reset will affect both timers.
		In single mode, reset will only affect TC0.
		The value of reading this bit will always be "0".

PSSR - Prescaler Select Register

PSSR - Prescaler Select Register

Address: 0xE2 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	PSS1	-	-	-	-	-	-	PSR1
R / W	R / W	-	-	-	-	-	-	W
Initial	0	0	0	0	0	0	0	0

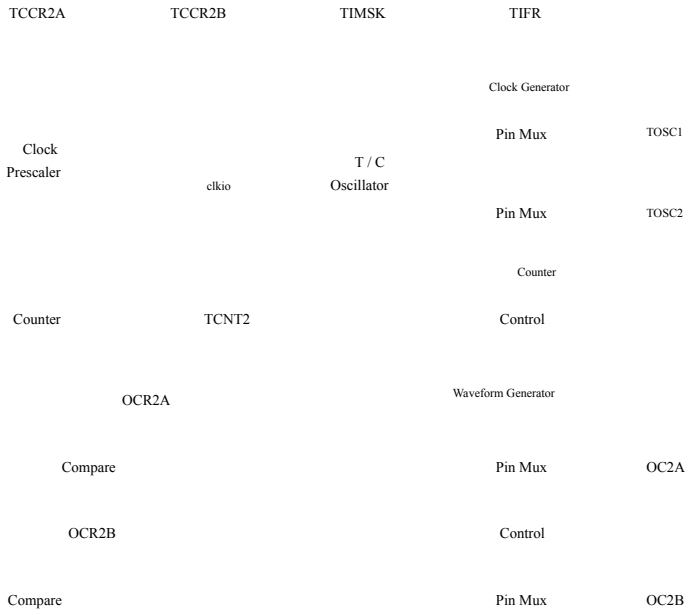
Bit	Name	Description
		Prescaler selection control bit.
		When the PSS1 bit is set to "1", it is the prescaler mode. TC0 alone use the prescaler CPS10,
7	PSS1	TC1 uses the prescaler CPS1 alone.
		When the PSS1 bit is set to "0", the prescaler is multiplexed. TC0 and TC1 share the prescaler
		CPS10. Prescaler CPS1 is invalid and will be reset.
6: 1	-	Keep it.
0	PSR1	Prescaler CPS1 reset control bit. The PSR1 bit is valid only in single mode.
		When the PSR1 bit is set to "1", the prescaler CPS1 will be reset. Hardware will be cleared after reset
		PSR1 bit. When setting the PSR1 bit to "0", the setting is invalid.
		The value of reading this bit will always be "0".

8 -bit timer / counter *2*

- 8-bit counter
- Two separate comparison units

- When the compare match occurs, the counter is automatically cleared and loaded automatically
- Phase-corrected PWM output without interference pulse
- Frequency generator
- External event counter
- 10-bit clock prescaler
- overflow and compare match interrupt
- Allows the use of external 32.768KHz RTC crystal count

Overview



TC2 structure diagram

TC2 is a general-purpose 8-bit timer counter module that supports PWM output and can produce waveforms precisely. TC2 package
It contains 1 8-bit counter, waveform generation mode control unit and 2 output comparison unit. Waveform generation mode control unit
Controls the operation mode of the counter and the generation of the comparison output waveform. Depending on the mode of operation, the counter is for each one
Count the clock Clkt2 to clear, add one or minus one operation. Clkt2 can be generated by an internal clock source or an external clock source.
When using an external 32.768KHz crystal count, TC2 can be used as an RTC counter. When the counter counts the value TCNT2

Reaches the maximum value (equal to the maximum value 0xFF or the output compare register OCR2A, defined as TOP, defining the maximum value for MAX
To distinguish between), the counter will be cleared or decremented. When the counter count value TCNT2 reaches the minimum value (etc.)
At 0x00, defined as BOTTOM), the counter will be added to an operation. When the counter count value TCNT2 arrives
OCR2A / OCR2B, also known as when a compare match occurs, will clear or set the output compare signal OC2A / OCR2B,
To generate PWM waveforms.

Operating mode

Timing counter 2 has four different modes of operation, including normal mode (Normal), compare match clear (CTC)
Mode, fast pulse width modulation (FPWM) mode and phase correction pulse width modulation (PCPWM) mode,
Generates the mode control bit WGM2 [2: 0] to select. The four modes are described below in detail. Because there are two independent output ratios
The units are represented by "A" and "B", respectively, and the lower output "x" is used to represent the two output compare cell channels.

Normal mode

The normal mode is the simplest mode of operation for the timer counter. The waveform generation mode control bit, WGM2 [2: 0] = 0,
The maximum value of the TOP is MAX (0xFF). In this mode, the count mode is incremented by one for each count clock
After the counter reaches the TOP overflow, it returns to BOTTOM to restart. In the same value as the count value TCNT2 becomes zero
Set the timer counter overflow flag TOV2 in the clock. The TOV2 flag in this mode is like the 9th count bit, just
Will only be set to not be cleared. The overflow interrupt service routine automatically clears the TOV2 flag, which the software can use to improve it
The resolution of the counter. There is no special case in normal mode to consider, you can always write a new count value.
The waveform of the output compare signal OC2x can be obtained by setting the data direction register of the OC2x pin to output. When COM2x = 1
, The OC2x signal is flipped when a compare match occurs. The frequency of the waveform in this case can be calculated using the following formula:

$f_{oc2xnormal} = f_{sys} / (2 * N * 256)$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).
The output compare unit can be used to generate interrupts, but interrupts are not recommended in normal mode, which takes too much CPU
time.

CTC mode

When the WGM2 [2: 0] = 2 is set, the timer counter 2 enters the CTC mode, and the maximum value of the count is OCR2A. in
In this mode, the count mode is incremented by one for each count clock. When the value of the counter TCNT2 equals TOP
The counter is cleared. OCR2A defines the maximum value of the count, that is, the resolution of the counter. This mode allows the user to be tolerant
Easy control matches the frequency of the output and also simplifies the operation of the external event count.
When the counter reaches the maximum value of the count, the output compare match flag OCF2 is set and the corresponding interrupt enable is set
Will be interrupted. The OCR2A register can be updated in the interrupt service routine to count the maximum value. In this mode
OCR2A does not use double buffering, the counter is updated with no prescaler or very low prescaler to maximum
Be careful when approaching the minimum. If the value written to OCR2A is less than the current TCNT2 value, the counter will be lost
One match match. Before the next match match occurs, the counter has to count to TOP before starting from BOTTOM
Start counting to OCR2A value. As with the normal mode, the count value is set back to the BOTTOM count clock to set the TOV2 flag
Mind The waveform of the output compare signal OC2x can be obtained by setting the data direction register of the OC2x pin to output. when
When COM2x = 1, the OC2x signal is flipped when a compare match occurs. In this case, the frequency of the waveform can be expressed by the following equation
To calculate:

$f_{oc2xcte} = f_{sys} / (2 * N * (1 + OCR2A))$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024). As can be seen from the formula, when setting OCR2x

For 0x0 and no prescaler, the output waveform with maximum frequency $f_{sys} / 2$ can be obtained .

Fast PWM mode

When setting WGM2 [2: 0] = 3 or 7, the timer counter 2 enters the fast PWM mode and can be used to generate high frequency
PWM waveform, the maximum value TOP is MAX (0xFF) or OCR2A, respectively. Fast PWM mode and other PWM
The pattern is different in that it is a one-way operation. The counter is incremented from the minimum value of 0x00 to TOP and then back to BOTTOM
number. When the count value TCNT2 reaches OCR2x or BOTTOM, the output compare signal OC2x is set or cleared.
Refer to compare the output mode COM2x settings, see the register description for details. As a result of one-way operation, fast PWM mode
The operating frequency is twice the phase-corrected PWM mode using bidirectional operation. High frequency characteristics make fast PWM mode
Suitable for power regulation, rectification and DAC applications. High-frequency signals can reduce the scale of external components (inductive capacitors, etc.)
So as to reduce system cost.
When the count value reaches the maximum value, the timer counter overflow flag TOV2 will be set and the value of the compare buffer will be updated
To the comparison value. If the interrupt is enabled, the compare buffer OCR2x register can be updated in the interrupt service routine.
The waveform of the output compare signal OC2x can be obtained by setting the data direction register of the OC2x pin to output. The frequency of the waveform
The rate can be calculated using the following formula:

$f_{oc2xfpwm} = f_{sys} / (N * (1 + TOP))$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).
When a compare match occurs between TCNT2 and OCR2x, the waveform generator sets (clears) the OC2x signal. When TCNT2 is cleared
At zero, the waveform generator clears (sets) the OC2x signal to generate a PWM wave. Whereby the extreme value of OCR2x will be
Will produce a special PWM waveform. When OCR2x is set to 0x00, the output PWM is every (1 + TOP) count clock
There is a narrow spike. When OCR2x is set to the maximum value, the output waveform is a continuous high or low level.

Phase correction PWM mode

When setting WGM2 [2: 0] = 1 or 5, the timer counter 2 enters the phase correction PWM mode, the maximum value of the count TOP is MAX (0xFF) or OCR2A, respectively. The counter is bidirectional, incremented by BOTTOM to TOP, and then again Decrements to BOTTOM, and repeats this operation. When the count reaches TOP and BOTTOM, the counting direction is changed and the count value is TOP or BOTTOM only stay on a count clock. In the process of increment or decrement, the count value TCNT2 and OCR2x When matching, the output compare signal OC2x will be cleared or set, depending on the setting of the compare output mode COM2x. With a single Compared to the operation, the maximum frequency available for bidirectional operation is small, but its excellent symmetry is more suitable for motor control. In phase correction PWM mode, the TOV2 flag is set when the count reaches BOTTOM. When the count reaches TOP, compare The value of the buffer is updated to the comparison value. If the interrupt is enabled, the compare buffer OCR2x can be updated in the interrupt service routine Deposit.

The waveform of the output compare signal OC2x can be obtained by setting the data direction register of the OC2x pin to output. The frequency of the waveform The rate can be calculated using the following formula:

$f_{oc2xpcpwm} = f_{sys} / (N * TOP * 2)$

Where N represents the prescaler factor (1, 8, 64, 256, or 1024).

During the count up, when the TCNT2 matches OCR2x, the waveform generator clears (sets) the OC2x signal. in In the process of decrementing the count, when the TCNT2 matches OCR2x, the waveform generator sets (clears) the OC2x signal. thus The extreme value of OCR2x will produce a special PWM wave. When the OCR2x is set to the maximum or minimum value, the OC2x signal is output Will remain low or high.

In order to ensure that the output PWM wave symmetry on both sides of the minimum value, in the absence of a comparison match, there are two cases Flip OC2x signal. The first case is when the value of OCR2x changes from the maximum value 0xFF to other data. When OCR2x Is the maximum value, the count value reaches the maximum, OC2x output and the previous descending count when the results of the same match, that is,

Hold OC2x unchanged. The value of the new OCR2x (non-0xFF) is updated and the value of OC2x is always maintained. When the ascending order counts, a compare match occurs. At this point the OC2x signal is not centered at the minimum, so it needs to be When TCNT2 reaches the maximum value, it flips the OC2x signal, which means that the first match of the OC2x signal is reversed when no compare match occurs condition. The second case is that when TCNT2 starts counting from a value higher than OCR2x, it will lose a compare match, Thus causing the occurrence of asymmetric situations. Also need to flip the OC2x signal to achieve the minimum on both sides of the symmetry.

TC2 asynchronous operation mode

When the AS2 bit in the ASSR register is "1", TC2 operates in asynchronous mode and the counter clock source is derived from Oscillator for external timer counter. Asynchronous mode TC2 operation to consider the following points.

- The transition between synchronous and asynchronous modes can cause TCNT2, OCR2A, OCR2B, TCCR2A and TCCR2B Data corruption. The safe operation steps are as follows:
  1. Clear the OCIE2A, TOIE2, and OCIE2B register bits to turn off TC2 interrupts;
  2. Set the AS2 bit to select the appropriate clock source;
  3. Write new data to TCNT2, OCR2A, TCCR2A, OCR2B and TCCR2B registers;
  4. When switching to asynchronous mode, wait for TCN2UB, OCR2AUB, TCR2AUB, OCR2BUB, and TCR2BUB Bit is cleared;
  5. Clear TC2 interrupt flag bit;
  6. Enable interrupts that need to be used.
- The oscillator is best to use 32.768KHz watch crystal. The system clock frequency must be four times higher than the crystal frequency.
- CPU write TCNT2, OCR2A, TCCR2A, OCR2B and TCCR2B, the hardware will be the first data into the scratchpad, The rising edge of the two TOSC1 clocks is latched into the corresponding register. The data is latched from the scratchpad to the destination store The new data write operation can not be performed before the device. Each register has its own independent scratchpad, so write TCNT2 And does not interfere with writing OCR2. The asynchronous status register, ASSR, is used to check whether the data has been written to the destination register.
- If TC2 is used as the wake-up condition for MCU sleep mode, it can not be entered before each register update ends Sleep mode, otherwise the MCU may enter the sleep mode before the TC2 setting takes effect, so the TC2 can not wake up the system System.
- If TC2 is used as the wake-up condition for MCU sleep mode, care must be taken to re-enter the sleep mode. in The interrupt logic requires a TOSC1 clock cycle to be reset if the time from wake-up to re-entry dormancy is less than one TOSC1 clock cycle, the interrupt will no longer occur, the device can not wake up. It is recommended to use the following methods:

1. Write the appropriate data for each register;
2. Wait for the corresponding update busy flag of ASSR to clear;
3. Go to sleep mode.

- If the asynchronous operating mode is selected, the TC2 oscillator will always operate unless it enters the power down mode. Users must be aware that, The stabilization time of this oscillator may be as long as 1 second, so it is recommended that the user wait at least after enabling the TC2 oscillator 1 second and then use TC2 asynchronous mode of operation.
- Asynchronous operation mode Sleep mode wake-up process: After the interrupt condition is met, the next timer clock starts to call Wake up the process. That is, the counter accumulates at least one clock before the processor can read the value of the counter. After the wake-up, the MCU executes the interrupt service routine and then executes the program after the SLEEP statement.
- Reading the value of TCNT2 within a short time after waking up from Sleep mode may return incorrect data. Because TCNT2 is Driven by an asynchronous TOSC1 clock, while reading TCNT2 must be synchronized via an internal system clock register To complete, the synchronization occurs on the rising edge of each TOSC1. After wake-up from Sleep mode, the system clock is reactivated and read The TCNT2 value is entered before the sleep mode until the next rising edge of the TOSC1 is updated. The phase of the TOSC1 is completely unpredictable when awakened from sleep mode, depending on the wake-up time. Therefore, read TCNT2 The recommended sequence of values is:

1. Write an arbitrary value to OCR2A or TCCR2A;
2. Wait for the corresponding update busy flag to be cleared;
3. Read TCNT2.

- In asynchronous mode, the synchronization of the interrupt flag requires 3 system clock cycles plus 1 timer period. In the MCU can be The counter accumulates at least one clock before reading the counter value that causes the interrupt flag to be set. Output the comparison letter The number of changes is synchronized with the timer clock, not the system clock.

TC2 prescaler

The TC2 prescaler's input clock is called clkt2s and is selected by the AS2 bit in the ASSR register to select the internal system clock clkio or external TOSC1 clock source, the default is connected with the system clock clkio. If AS2 is set, TC2 will be set by TOSC1 Asynchronous drive. When the TOSC1 pin and TOSC2 pin external a 32.768KHz watch crystal, TC2 can be used as RTC meter The number of devices. It is not recommended to apply an external clock signal directly to the TOSC1 pin.

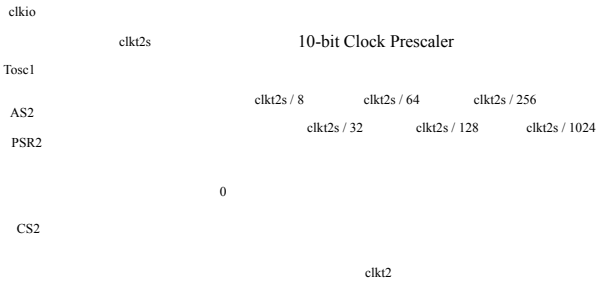


Figure 5 TC2 prescaler structure

The figure shows the TC2 prescaler, as shown in the figure, the possible prescaler options are: clkt2s / 8, clkt2s / 32, clkt2s / 64, clkt2s / 128, clkt2s / 256, and clkt2s / 1024. You can also select clkt2s and 0 (stop counting). Set the PSR2 of the SFIOR register Bit will reset the prescaler, allowing the user to start working from a predictable prescaler.

Register definition

TC2 register list

register	address	Defaults	description
TCCR2A	0xB0	0x00	TC2 control register A
TCCR2B	0xB1	0x00	TC2 control register B

TCNT2	0xB2	0x00	TC2 count register
OCR2A	0xB3	0x00	TC2 Output Compare Register A
OCR2B	0xB4	0x00	TC2 Output Compare Register B
ASSR	0xB6	0x00	TC2 Asynchronous Status Register
TIMSK2	0x70	0x00	The timer counter interrupts the mask register
TIFR2	0x37	0x00	Timer counter interrupt flag register

TCCR2 A-TC2 Control Register A

TCCR2 A - TC2 Control Register A

Address: 0xB0 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20
R / W	W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	COM2A1	TC2 compare match output A mode control high. COM2A1 and COM2A0 together constitute the output compare mode control COM2A [1: 0], control OC2A Output waveform. If the 1 or 2 bits of COM2A are set, the output compare waveform occupies OC2A Pin, but the pin's data direction register must be set high to output this waveform. In different work Mode, COM2A on the output comparison waveform control is also different, see the specific comparison output mode control Table description.
6	COM2A0	TC2 compare match output A mode control low. COM2A0 and COM2A1 together constitute the output compare mode control COM2A [1: 0], control OC2A Output waveform. If the 1 or 2 bits of COM2A are set, the output compare waveform occupies OC2A Pin, but the pin's data direction register must be set high to output this waveform. In different work Mode, COM2A on the output comparison waveform control is also different, see the specific comparison output mode control Table description.
5	COM2B1	TC2 compare match output B mode control high. COM2B1 and COM2B0 together form the output compare mode control COM2B [1: 0], which controls the OC2B Output waveform. If the 1 or 2 bits of COM2B are set, the output compare waveform occupies OC2B Pin, but the pin's data direction register must be set high to output this waveform. In different work Mode, COM2B on the output comparison waveform control is also different, see the specific comparison output mode control Table description.
4	COM2B0	TC2 compare match output B mode control low. COM2B0 and COM2B1 together constitute the output compare mode control COM2B [1: 0], control OC2B Output waveform. If the 1 or 2 bits of COM2B are set, the output compare waveform occupies OC2B Pin, but the pin's data direction register must be set high to output this waveform. In different work Mode, COM2B on the output comparison waveform control is also different, see the specific comparison output mode control Table description.
3: 2	-	Keep it.
1	WGM21	TC2 waveform generation mode control high. WGM20 and WGM21, WGM22 together form the waveform generation mode control WGM2 [2: 0], control Counter count mode and waveform generation mode, see the waveform generation mode table description.
0	WGM20	TC2 waveform generation mode control low. WGM21 and WGM20, WGM22 together form the waveform generation mode control WGM2 [2: 0], control Counter count mode and waveform generation mode, see the waveform generation mode table description.

Page 128

TCCR2B -TC2 Control Register B

TCCR2B - TC2 Control Register B

Address: 0xB1 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20
R / W	W	W	-	-	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	FOC2A	TC2 Force Output Compare A control bit. When operating in non-PWM mode, it can be done by writing "1" to the forced output compare bit FOC2A Compare match. Forcing the compare match does not set the OCF2A flag, nor does it override or clear the timer, But the output pin OC2A will be updated according to the COM2A settings accordingly, just what happens Match the same. The return value of reading FOC2A is always zero.
6	FOC2B	TC2 Force Output Compare B control bits. When operating in non-PWM mode, it can be done by writing "1" to the forced output compare bit FOC2B Compare match. Forcing the compare match does not set the OCF2B flag, nor does it override or clear the timer, But the output pin OC2B will be updated according to the COM2B settings accordingly, just as it really happens Match the same. The return value of the read FOC2B is always zero.
5: 4	-	Keep it.
3	WGM22	TC2 waveform generation mode control high. WGM22 and WGM20, WGM21 together form the waveform generation mode control WGM2 [2: 0], the control meter Count counting method and waveform generation mode, see the waveform generation mode table description.
2	CS22	TC2 clock selection control high. Used to select the clock source for Timer 2.
1	CS21	TC2 clock selection control bit. Used to select the clock source for Timer 2.
0	CS20	TC2 clock select control low. Used to select the clock source for Timer 2.

CS2 [2: 0]	description
0	No clock source, stop counting
1	clk $_{t2s}$
2	clk $_{t2s}$ / 8, from prescaler
3	clk $_{t2s}$ / 32, from prescaler
4	clk $_{t2s}$ / 64, from prescaler
5	clk $_{t2s}$ / 128, from prescaler
6	clk $_{t2s}$ / 256, from prescaler
7	clk $_{t2s}$ / 1024, from prescaler

Page 129

The following table shows the non-PWM mode (ie, normal mode and CTC mode), compare the output mode to the output compare waveform control.

Table 2 OC2x Compare Output Mode Control in Non-PWM Mode

COM2x [1: 0]	description
0	OC2x disconnect, general IO port operation
1	Match the OC2x signal when comparing the match
2	The OC2x signal is cleared when compare match
3	The OC2x signal is set when the compare match

The following table shows the control of the output compare waveform for the compare output mode in fast PWM mode.

Table 3 OC2x Compare Output Mode Control in Fast PWM Mode

COM2x [1: 0]	description
0	OC2x disconnect, general IO port operation
1	Keep it
2	The OC2x signal is cleared when the compare match is set and the OC2x signal is set when the maximum match is made
3	The OC2x signal is set when the compare match is cleared and the OC2x signal is cleared when the maximum match is made

The following table shows the control of the output compare waveform for the compare output mode in phase correction mode.

Table 4 Phase Correction PWM Mode OC2x Compare Output Mode Control

COM2x [1: 0]	description
0	OC2x disconnect, general IO port operation
1	Keep it
2	The OC2x signal is cleared when the compare match is cleared in the ascending count, and the OC2x signal is set at the compare match
3	The OC2x signal is set when the compare match is set in the ascending count, and the OC2x signal is cleared when the compare match is cleared

The following table shows the waveform generation mode control.

Table 5 Waveform Generation Mode Control

WGM2 [2: 0]	Operating mode	TOP value	Update OCR2x at all times	Set TOV2 at all times
0	Normal	0xFF	immediately	MAX
1	PCPWM	0xFF	TOP	BOTTOM
2	CTC	OCR2A	immediately	MAX
3	FPWM	0xFF	TOP	MAX
4	Keep it	-	-	-
5	PCPWM	OCR2A	TOP	BOTTOM
6	Keep it	-	-	-
7	FPWM	OCR2A	TOP	TOP

TCNT2 -TC2 count value register

TCNT2 - TC2 count register

Address: 0xB2 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	TCNT27	TCNT25	TCNT25	TCNT23	TCNT23	TCNT21	TCNT21	TCNT20
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	TCNT2	TC2 count register. Through the TCNT2 register can be directly on the counter 8 for the count value to read and write access. The CPU writes to the TCNT2 register will prevent the compare match from occurring during the next timer clock cycle Health, even if the timer has stopped. This allows the value of the TCNT2 register to be initialized with the value of OCR2 So that no interruptions are caused. If the value written to TCNT2 equals or bypasses the OCR2 value, the comparison match is lost, resulting in incorrect The resulting waveform results. The timer stops counting when no clock source is selected, but the CPU can still access TCNT2. CPU write counter Has a higher priority than zeroing or addition and subtraction.

OCR2A - TC2 Output Compare Register **A**

OCR2A - **TC2** Output Compare Register **A**

Address: 0xB3 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	OCR2A7	OCR2A6	OCR2A5	OCR2A4	OCR2A3	OCR2A2	OCR2A1	OCR2A0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	OCR2A	TC2 Output Compare Register A. The OCR2A contains an 8-bit data that is continuously compared with the counter value TCNT2. Compare match Can be used to generate an output compare interrupt, or to generate a waveform on the OC2A pin. When using PWM mode, the OCR2A register uses a double buffered register. While the normal working mode and match With clear mode, double buffering is disabled. Double buffering can be updated with OCR2A register with count Maximum or minimum moments to prevent asymmetrical PWM pulses from being generated, eliminating interference pulse. When using the double buffering function, the CPU accesses the OCR2A buffer register, when the dual buffering function is disabled. Asked about OCR2A itself.

OCR2B - TC2 Output Compare Register **B**

OCR2B - **TC2** Output Compare Register **B**

Address: 0xB4 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	OCR2B7	OCR2B6	OCR2B5	OCR2B4	OCR2B3	OCR2B2	OCR2B1	OCR2B0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	OCR2B	TC2 Output Compare B register. The OCR2B contains an 8-bit data that is continuously compared with the counter value TCNT2. Compare horses Can be used to generate an output compare interrupt, or to generate a waveform on the OC2B pin. When using PWM mode, the OCR2B register uses a double buffered register. While the normal working mode and match With clear mode, double buffering is disabled. Double buffering can be updated with OCR2B registers with count Maximum or minimum moments, thereby preventing the generation of asymmetric PWM pulses, eliminating the need for dry Disturbing pulses. When using the double buffering function, the CPU accesses the OCR2B buffer register, which inhibits double buffering

When the CPU can access the OCR2B itself.

TIMSK2 - TC2 interrupt mask register

TIMSK2 - TC2 interrupt mask register

Address: 0x70 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2
R / W	-	-	-	-	-	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 3		Keep it.
2	OCIE2B	TC2 Output Compare B Match Interrupt Enable bit. When the OCIE2B bit is set to "1" and the global interrupt is set, the TC2 output compares the B match interrupt enable. When the match is matched When an OCF2B bit is set in TIFR2, an interrupt is generated. When the OCIE2B bit is "0", the TC2 output compare B match interrupt is disabled.
1	OCIE2A	TC2 Output Compare A Match Interrupt Enable bit. When the OCIE2A bit is "1" and the global interrupt is set, the TC2 output compares the A match interrupt enable. When the match is matched Occurs when the OCF2A bit in TIFR2 is set, the interrupt is generated. When the OCIE2A bit is "0", the TC2 output compare A match interrupt is disabled.
0	TOIE2	TC2 overflow interrupt enable bit. When the TOIE2 bit is "1" and the global interrupt is set, the TC2 overflow interrupt is enabled. When TC2 overflows, ie TIFR2 When the TOV2 bit is set, the interrupt is generated. When the TOIE2 bit is "0", the TC2 overflow interrupt is disabled.

TIFR2 - TC2 interrupt flag register

TIFR2 - TC2 interrupt flag register

Address: 0x37 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	OCF2B	OCF2A	TOV2
R / W	-	-	-	-	-	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

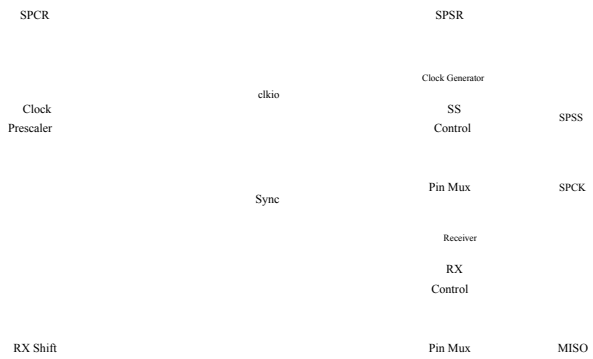
Bit	Name	description
7: 3	-	Keep it.
2	OCF2B	TC2 Output Compare B match flag. When TCNT2 is equal to OCR2B, the compare unit gives the match signal and sets the compare flag OCF2B. If At this time, the output compare B interrupt enable OCIE2B to "1" and the global interrupt flag is set, the output ratio is generated More than B interrupted. OCF2B is automatically cleared when writing to this interrupt service routine, or "1" for OCF2B bit This bit can be cleared.
1	OCF2A	TC2 Output Compare A match flag. When TCNT2 is equal to OCR2A, the compare unit gives the match signal and sets the compare flag OCF2A. If At this time, the output compare A interrupt enable OCIE2A to "1" and the global interrupt flag is set, the output ratio is generated Than A interrupt. OCF2A is automatically cleared when writing to this interrupt service routine, or "1" for OCF2A bit This bit can be cleared.
0	TOV2	TC2 overflow flag. When the counter overflows, set the overflow flag TOV2. If the overflow interrupt enable TOIE2 is "1" And the global interrupt flag is set, an overflow interrupt is generated. TOV2 is automatically executed when this interrupt service routine is executed Clear this bit or write "1" to the TOV2 bit.

SPI serial peripheral interface

- Full-duplex, three-wire synchronous data transmission
- Host or slave operation
- The least significant or most advanced bit is transmitted first
- 7 programmable bit rates
- Send end interrupt flag
- Write conflict flag protection mechanism
- wake-up from idle mode
- The host operation has speed mode
- Supports host two-wire input mode

Summary

SPI mainly consists of three parts: clock prescaler, clock detector, slave select detector, transmitter and receiver
Device. The control and status registers are shared by these three parts. The clock prescaler works only in the master operating mode by bit
Rate control bit to select the division factor, resulting in the corresponding frequency division clock, the output to the SPCK pin. Clock detector only
Work in slave operation mode, detect the clock edge input from the SPCK pin, according to the SPI data transfer mode
Send and receive shift register for shift operation. The slave selection detector detects the slave selection signal SPSS
Transmission status to control the operation of the transmitter and receiver. The transmitter consists of a shift register and transmit control logic.
The receiver consists of a shift register, a receive buffer, and receive control logic.



SPI structure diagram

Clock generation

Clock generation logic is divided into host clock prescaler and slave clock detector, respectively, in the host operation and from the machine Mode. The clock prescaler selects the division factor from the bit rate control bit and the speed control bit, resulting in the corresponding frequency division Clock (a total of 7 kinds of optional frequency division coefficient, see the register description for details), output to the SPCK pin for communication Clock, while providing a shift clock for the internal transmit and receive shift registers. The clock detector performs edge on the input clock SPCK Along the detection, according to the SPI data transfer mode on the transmitter and receiver shift operation. To ensure positive for the clock signal Sampling, SPCK clock high and low width must be greater than 2 system clock cycles.

Send and receive

The SPI module supports both simultaneous transmission and reception in single-wire mode, and only supports dual-line reception in dual-line mode.

Single line send and receive

SPI host will need to communicate from the slave select signal SPSS pulled low, you can start a transmission process. Host and slave Will need to transfer the data ready, the host clock signal SPCK generate clock pulses to exchange data, the host data Removed from the MOSI, moved from the MISO, the slave data from the MISO removed, moved from the MOSI, after the exchange of data after the host Pull the SPSS signal to complete the communication.

When configured as a master, the SPI module does not control the SPSS pin and must be handled by the user software. Software pulls down SPSS Pin, select the slave to communicate, start the transmission. The software will need to transfer the data to the SPDR register when it is started Clock generator, the hardware generates the clock of the communication, and moves the 8-bit data out of the slave, while moving the slave data into the slave. Shift After a byte of data, the clock generator is stopped and the transmit completion flag SPIF is set. The software can write the data again SPDR register to continue transmitting the next byte, or you can pull the SPSS signal to end the current transfer. The last number Will be stored in the receive buffer.

When configured as a slave, the SPI module will remain asleep as long as the SPSS signal is always high and keep the MISO Feet for the three states. The software can update the contents of the SPDR register. Even if there is an input clock pulse on the SPCK pin at this time, SPDR data will not be removed until the SPSS signal is pulled low. When a byte of data is transferred, the hardware is done Set the transmission completion flag SPIF. At this point the software to read the data before moving to the SPDR register to continue to write data, The last incoming data will be stored in the receive buffer.

The SPI module has only one buffer in the transmit direction and two buffers in the receive direction. When sending data, one Wait until the shift process is complete before writing to the SPDR register. And in the receiving data, the need for The next byte before the end of the shift process by accessing the SPDR register to read the received characters, otherwise the previous word Section will be lost.

Host two-wire reception

SPI module two-line mode is only effective in the host operating mode, and the single-line mode is different MOSI and MISO are For the host to receive data, each SPCK clock pulse simultaneously receives 2 bits of data (the data on the MISO line is Before the data on the MOSI line is on), after receiving the two bytes of data, the hardware sets the transmission completion flag SPIF, the data Saved to the receive buffer and shift register. At this point the software must read the SPDR register twice to get the two received

Byte of data. It should be noted that, although the two-line mode, the host does not send data to the slave, the software still need to SPDR Register write data to start the clock generator to generate the communication clock, write once the SPDR register to receive two bytes

The data.

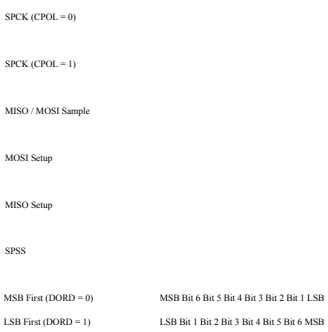
Data pattern

Single-line mode, SPI relative to the serial data, there are four SPCK phase and polarity combination, by the CPHA and CPOL To control, as shown in the following table.

CPHA and CPOL Select the data transfer mode

CPOL	CPHA	Starting along	End the edge	SPI mode
0	0	Sampling (rising edge)	Set (falling edge)	0
0	1	Set (rising edge)	Sampling (falling edge)	1
1	0	Sampling (falling edge)	Set (rising edge)	2
1	1	Set (falling edge)	Sampling (rising edge)	3

When CPHA = 0, the data samples and set the clock edge as shown below:



CPHA is "0" when SPI data transfer mode

When CPHA = 1, the data samples and set the clock edge as shown below:

CPHA is "1" when SPI data transfer mode

Two-wire mode, MISO and MISO are used as host input, the data sampling time is still determined by the data transmission mode,

Sampling as shown below:

MISO / MOSI Sample								
MSB First (DORD = 0)								
MISO	MSB	Bit 5	Bit 3	Bit 1	MSB	Bit 5	Bit 3	Bit 1
MOSI	Bit 6	Bit 4	Bit 2	LSB	Bit 6	Bit 4	Bit 2	LSB
LSB First (DORD = 1)								
MISO	LSB	Bit 2	Bit 4	Bit 6	LSB	Bit 2	Bit 4	Bit 6
MOSI	Bit 1	Bit 3	Bit 5	MSB	Bit 1	Bit 3	Bit 5	MSB

Bit mode when the DUAL is "1" when the SPI data sampling mode

SPSS pin function

When configured as a slave, the slave select signal SPSS pin is always used as input. When the SPSS pin remains low, the SPI is connected
The port is activated, the MISO pin becomes the output pin (the software performs the corresponding port configuration), and the other pins are input. When SPSS
When the pin is held high, the SPI module is reset and no longer receives data. The SPSS pin is very synchronous for packet / byte synchronization
It is useful to synchronize the slave's bit counter with the host's clock generator. When SPSS is pulled high, the SPI slave is immediately reset
Receive and send logic, and discard the incomplete data in the shift register.

When configured as a host, the user software can determine the direction of the SPSS pin.
If SPSS is configured as an output, it can be used to drive the slave's SPSS pin. If SPSS is configured as an input, it must be maintained
High to ensure the normal work of the host. When configured as a master and the SPSS pin is an input, the external circuit pulls down the SPSS pin,
The SPI module will consider another host to select itself as a slave and start transmitting data. To prevent bus conflicts, SPI
The module will do the following:

1. Clear the MSTR bit in the SPCR register, convert to slave, so MOSI and SPCK become input;
2. Set the SPIF bit in the SPSR register. If the interrupt is enabled, an SPI interrupt is generated.

Therefore, when interrupts are used to process the data transfer of the SPI host and there is a possibility that the SPSS is pulled down, the interrupt service
The program should check whether the MSTR bit is "1". If cleared, the software must set it to re-enable SPI master mode.

SPI initialization

The SPI is first initialized before communication is performed. The initialization process usually includes the choice of host slave operation, data transfer
The setting of the input mode, the selection of the bit rate, and the direction control of the respective pins. Where the host and slave operate under the pin side
The controls are different, as shown in the following table:

Pin direction control

Pin	The direction of the host mode	Slave mode direction
MOSI	User software definition	enter
MISO	enter	User software definition
SPCK	User software definition	enter
SPSS	User software definition	enter

SPI host initialization

The SPI host mode initialization process is as follows:

1. Set the MSTR bit, set the bit rate selection control bit, data transfer mode, data transfer order, interrupt enable or not,
   As well as double-line enabled or not;
2. Set the MOSI and SPCK pins to output;

3. Set the SPE bit.

In host mode, the SPSS pin can be set to output when the SPI module is not desired to be used as a slave by another host.

SPI slave initialization

The SPI slave mode initialization process is as follows:

1. Clear the MSTR bit, set the data transfer mode, data transfer order, interrupt enable or not;
2. Set the MISO pin to output;
3. Set the SPE bit.

Register definition

SPI register list

register	address	Defaults	description
SPCR	0x4C	0x00	SPI control register
SPSR	0x4D	0x00	SPI status register
SPDR	0x4E	0x00	SPI data register

SPCR - SPI control register

***SPCR* - SPI** control register

Address: 0x4C Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	Description
7	SPIE	SPI interrupt enable bit. The SPI interrupt is enabled when the SPIE bit is set to "1". The SPIF bit located in the SPSR register is set And the global interrupt is enabled, an SPI interrupt is generated. When the SPIE bit is set to "0", the SPI interrupt is disabled.
6	SPE	SPI enable bit. The SPI module is enabled when the SPE bit is set to "1". The SPE must be set before any SPI operation. When the SPE bit is set to "0", the SPI module is disabled.
5	DORD	Data order control bit. When the DORD bit is set to "1", the LSB of the data is transmitted first.

		When the DORD bit is set to "0", the MSB of the data is transmitted first.
4	MSTR	Host Slave Selects the control bit. When the MSTR bit is set to "1", the host mode is selected. When the MSTR bit is set to "0", the slave mode is selected. In master mode, when the SPSS pin is configured as an input and is pulled low, the MSTR bit is cleared to the SPSR register The SPIF of the device is set and the user must reset the MSTR into host mode.
3	CPOL	Clock polarity control bit. When the CPOL bit is set to "1", SPCK is high in the idle state. When the CPOL bit is set to "0", SPCK is low in the idle state.

CPOL	Starting along	End the edge
0	Rising edge	Falling edge
1	Falling edge	Rising edge

2 CPHA clock phase control bit.

When the CPHA bit is set to "1", the start edge sets the data and ends the edge of the sampled data.

When the CPHA bit is set to "0", the start edge is sampled and the edge edge sets the data.

CPHA	Starting along	End the edge
0	sampling	Set up
1	Set up	sampling

1 SPR1 clock rate selection bit 1.

SPR1 and SPR0 are used to select the SPI transfer clock rate. See the SPCK and the system clock for specific control Form form.

0 SPR0 Clock Rate Select bit 0.

SPR1 and SPR0 are used to select the SPI transfer clock rate. See the SPCK and the system clock for specific control Form form.

SPSR - SPI status register

SPSR - **SPI** status register

Address: 0x4D Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	-	-	-	DUAL	-	SPI2X
R / W	R	R	R	R	R	R / W	R	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	SPIF	SPI interrupt flag. After the serial transmission is set, the SPIF flag is set. In the host mode, when the SPSS pin is configured as input and pulled low, SPIF will also be set. If the SPTER bit and the global interrupt enable bit of the SPCR register are set, the SPI is set Interrupt generated. After entering the interrupt service routine, the SPIF bit is automatically cleared, or by first reading the SPSR register Access the SPDR register to clear the SPIF bit.
6	WCOL	Write conflicting flag. Writing to the SPDR register during data transfer will set the WCOL bit. The WCOL bit can be read by first

		The SPSR register then accesses the SPDR register to clear.
5	-	Keep it.
4	-	Keep it.
3	-	Keep it.
2	DUAL	Two-wire mode control bit. When the DUAL bit is set to "1", the SPI 2-wire transmission mode is enabled. When the DUAL bit is set to "0", the SPI 2-wire transfer mode is disabled. The two-wire transfer mode is valid only in SPI master mode. Both MISO and MOSI are used as host data inputs. According to the transmission mode see the host dual-line reception and data mode section description.
1	-	Keep it.
0	SPI2X	SPI speed control bit. When the SPI2X bit is set to "1", the SPI transmission speed is doubled. When the SPI2X bit is set to "0", the SPI transmission speed is not doubled. Refer to SPCK and the system clock for the specific control mode.

The following table shows the relationship between SPCK and the system clock.

The relationship between SPCK and the system clock

SPI2X	SPR1	SPR0	SPCK frequency
0	0	0	f_{sys} / 4
0	0	1	f_{sys} / 16
0	1	0	f_{sys} / 64

0	1	1	f_{sys} / 128
1	0	0	f_{sys} / 2
1	0	1	f_{sys} / 8
1	1	0	f_{sys} / 32
1	1	1	f_{sys} / 64

SPDR - SPI data register

SPDR - **SPI** data register

Address: 0x4E Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	SPDR7	SPDR6	SPDR5	SPDR4	SPDR3	SPDR2	SPDR1	SPDR0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit Name	description
7: 0 SPDR	SPI sends and receives data. SPI sends data and receives data to share SPI data register SPDR. Writing data to SPDR is written to send The data shift register reads the data from the SPDR that reads the receive data buffer.

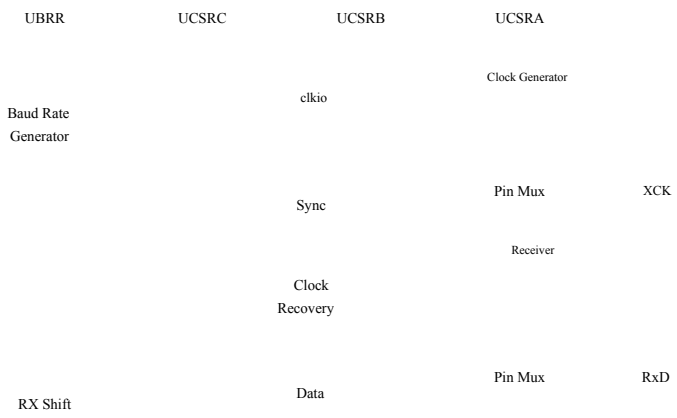
USART0 - Universal Synchronous / Asynchronous Serial Transceiver

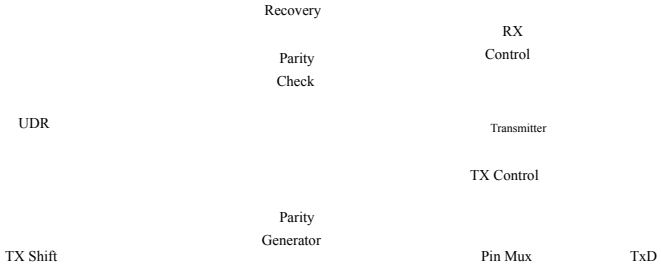
- Full-duplex operation (independent serial receive and transmit registers)
- Asynchronous or synchronous operation
- Host or slave operation

High-precision baud rate generator

- Supports 5, 6, 7, 8, or 9 data bits and 1, or 2 stop bits
- Hardware-enabled parity generation and verification mechanisms
- Data overspeed detection
- Frame error detection
- Noise filtering, including erroneous start bit detection and digital low-pass filter
- Three independent interrupts: Transmit end interrupt, transmit data register empty interrupt, and receive end interrupt
- Multi-processor communication mode
- Asynchronous communication mode

Summary





USART structure diagram

The USART mainly consists of three parts: a clock generator, a transmitter and a receiver. The control and status registers are made up of these three parts shared. The clock generator consists of the synchronization logic of the external input clock from the baud rate generator and the synchronous slave operating mode.

The XCK pin is used only for synchronous transfer mode. The transmitter includes a write data buffer, a serial shift register, and a parity occurs And the control logic required to handle different frame formats. The write data buffer allows continuous transmission of data without being in the data frame Between the introduction of delay. The receiver has a clock and data recovery unit for reception of asynchronous data. In addition to the recovery unit, receive The device also includes parity, control logic, serial shift register and a two-stage receive buffer UDR. Receiver support with The transmitter has the same frame format, and can detect frame errors, data overspeed and parity errors.

Clock generation

The clock generation logic generates the base clock for the transmitter and receiver. The USART supports four modes of clock: normal asynchronous mode Type, speed asynchronous mode, host synchronization mode, and slave synchronization mode. The UCREL bit of USCRC is used to select the same Step or asynchronous mode. The U2X bit of USCRA controls the speed enable in asynchronous mode. Valid only in synchronous mode XCK The data direction register (multiplexed with IO) of the pin determines whether the clock source is generated internally (host mode) or externally (from Machine mode).

Baud rate generator

The baud rate register UBRR and the down-count counter are connected together as a programmable prescaler or baud rate for the USART Generator. The descending counter operates under the system clock (f_{sys}) and is automatically incremented when its count is zero or the UBRRL register is written The value of the UBRR register is set. When the count to zero to produce a clock, the clock as the baud rate generator output clock, The frequency is f_{sys} / (UBRR + 1).

The following table shows the formulas for calculating baud rates (bits per second) and UBRR values in various operating modes.

Operating mode	Baud rate calculation formula (1)	**UBRR** value calculation formula
Asynchronous normal mode	BAUD = f_{sys} / (16 * (UBRR + 1))	UBRR = f_{sys} / (16 * BAUD) - 1
Asynchronous speed mode	BAUD = f_{sys} / (8 * (UBRR + 1))	UBRR = f_{sys} / (8 * BAUD) - 1
Synchronous host mode	BAUD = f_{sys} / (2 * (UBRR + 1))	UBRR = f_{sys} / (2 * BAUD) - 1

Description:

1. The baud rate is defined as the bit transfer rate (bps) per second;
2. BUAD is the baud rate, f_{sys} is the system clock, and UBRR is the combination value of the baud rate register UBRRH and UBRRL.

Speed mode of operation

By setting the U2RA bit in the UCSRA register, it is possible to double the transfer rate, which is valid only in asynchronous mode Set this bit to "0" in step mode.
Setting this bit will halve the division of the baud rate divider, effectively doubling the transfer rate of asynchronous communication. In this case , The receiver uses only half the number of samples to sample and clock the data, requiring a more accurate baud rate Set and system clock. The transmitter does not change.

External clock

The synchronous slave operating mode is driven by an external clock. The external clock passes through the sync register and the edge detector before being sent by the transmitter And the receiver is used, this process will introduce the delay of the two system clocks, so the maximum external clock frequency of the external XCK is given by Formula Restrictions:

$$f_{XCK} < f_{sys} / 4$$

Note that fsys has a system clock stability decision, in order to prevent data loss due to frequency drift, it is recommended to keep enough margin the amount.

Synchronous clock operation

In synchronous mode, the XCK pin is used for clock input (slave mode) or clock output (master mode). Clock edge with The basic rules governing the relationship between data sampling and data variation are the clock edge used for data input (RxD) sampling and data The clock edge used by the output changes is reversed.

UCPOL = 1

UCPOL = 0

XCK timing in synchronous mode

As shown in the figure above, when the UCPOL value is "1", the data output is changed on the falling edge of XCK, and on the rising edge of XCK Data is sampled; when the UCPOL value is "0", the data output is changed on the rising edge of XCK, and the number of XCK According to sampling.

Frame format

A serial data frame consists of a data word plus a sync bit (start bit and stop bit) and a parity bit for error correction.

The USART accepts the following 30 combinations of data frame formats:

- 1 start bit
- 5,6,7,8 or 9 data bits
- No parity, odd parity or even parity
- 1 or 2 stop bits

The data frame starts at the start bit, followed by the least significant bit of the data word, followed by the other data bits, with the highest bit of the data word Beam, the most successful transmission of 9-bit data. If parity is enabled, the parity bit will be followed by the data word, and finally the stop bit. when After a complete data frame is transmitted, the next new data frame can be transmitted immediately or the transmission line is idle (high Flat state). The following figure shows the possible data frame structure, and the bits in square brackets are optional.

USART frame structure

Description:

Page 143

1) There is no data transmission on the IDLE communication line (RxD or TxD) and must be high when the line is idle

2) St start bit, always low

3) 0-8 data bits

4) P parity, odd parity or even parity

5) Sp stop bit, always high

The structure of the data frame is set by UCSZB and UCSZ [2: 0], UPM [1: 0] and USBS in the UCSRB register. Receive and send
Send using the same settings. Any changes made may destroy the ongoing data transfer. Among them, UCSZ [2: 0] is determined
The data bits of the data frame, UPM [1: 0] is used to enable and determine the type of parity, USBS sets the frame with one or two bits
Beam position. The receiver will ignore the second stop bit, so the frame error is detected only when the first end bit is "0".

Check bit calculation

The parity bit is calculated by XORing the bits of the data. If odd parity is selected, the XOR result needs to be reversed.
The relationship between the parity bit and the data bits is as follows:

$$P_{\text{the even}} = D_{n-.1} \oplus \ldots \oplus d_{.3} \oplus d_{2} \oplus d_{.1} \oplus d_{0} \oplus 0$$

$$P_{\text{ODD}} = D_{n-.1} \oplus \ldots \oplus d_{.3} \oplus d_{2} \oplus d_{.1} \oplus d_{0} \oplus 1$$

Description:

1) P_{even} Even check results

2) P_{odd} odd parity results

3) d_{n} nth data bits

The USART is initialized

The USART should be initialized before communication. The initialization process usually includes the baud rate setting, the frame structure
Set, and enable receivers or transmitters as needed. For interrupt-driven USART operations, clear during initialization
Zero global interrupt flag and disables all interrupts for the USART.
When performing a re-initialization such as changing the baud rate or frame structure, it is necessary to ensure that no data is transmitted. TXC flag can be used
To detect whether the transmitter has completed all transmissions, the RXC flag can be used to detect whether there is data in the receive buffer
Read out. If the TXC flag is used for this purpose, it must be cleared before each data is sent (before writing the UDR register)
Zero TXC flag.

Transmitter

Setting the TXEN bit of the UCSRB register will enable data transmission for the USART. Enable the general IO function of the TxD pin
Replaced by the USART function, becomes the serial output of the transmitter. Before sending data to set the baud rate, work mode
With frame format. If the synchronous transmit mode is used, the clock signal applied to the XCK pin is the clock for data transmission.

Send 5 to 8 frames for data

Will need to send the data loaded into the send buffer to start the data sent. The CPU loads the number by writing to the UDR register
according to. When the transmit register can send a new frame of data, the data in the buffer will be transferred to the shift register.
When the shift register is in the idle state (no ongoing data transfer), or the last stop of the previous frame data
When the bit is sent, it will load the new data. Once the shift register has loaded the new data, it will pass in the established settings
Lose a complete frame.

Page 144

A frame that sends 9-bit data

If the 9-bit data frame is sent, the ninth bit of the data should be written to the TXB8 bit in register UCSRB and then low
The bit data is written to the transmit data register UDR. The 9th bit data is used in the multi-machine communication to represent the address frame in synchronous communication
Can be used for protocol processing.

Send parity bit

The parity generation circuit generates the corresponding parity bit for the serial data frame. When the parity bit is enabled (UPM1 = 1), send control
The logic circuit inserts the parity bit between the last bit of the data word and the first stop bit.

Send flag and interrupt processing

The USART transmitter has two flags: the USART data register empty flag UDRE and the transmission end flag TXC, two
Ambitions can be interrupted.
The data register empty flag UDRE is used to indicate whether the transmit buffer can write a new data. The bit is in the send buffer
When the space is set to "1", the full time is set to "0". When the UDRE bit is "1", the CPU can write to the data register UDR
New data, and vice versa.
When the data register empty interrupt enable bit UDRIE in the UCSRB register is "1", as long as the UDRE is set (and all
Disable), the USART data register will generate an empty interrupt request. Writing to the register UDR will be cleared
UDRE. When transmitting data in an interrupt mode, a new number must be written in the data register empty interrupt service routine
To the UDR to clear the UDRE, or to disable the data register from being interrupted. Otherwise once the interrupt service routine ends,
A new interrupt will be generated again.
When the entire data frame is shifted out of the transmit shift register and there is no new data in the transmit register, the transmit end flag
TXC will be set. When the transmit end interrupt enable bit, TXCIE (and global interrupt enable) on UCSRB, is set to "1",
With the TXC flag set, the USART transmit end interrupt will be executed. Once the interrupt service routine is entered, the TXC flag bit
It is automatically cleared, the CPU can also write "1" to the bit to clear.

Disable the transmitter

When TXEN is cleared, only after all the data have been sent to send the transmitter can be really prohibited, that is, send shift send
There is no data to be transferred in both the register and the transmit buffer register. After the transmitter is disabled, the TxD pin resumes its general purpose IO work
can.

receiver

Set the receive enable bit (RXEN) of the UCSRB register to start the USART receiver. Enabled after the RxD pin is generic
The IO function is replaced by the USART function as the serial input of the receiver. Be sure to set up the data before receiving it
Baud rate, operating mode and frame format. If the synchronous receive mode is used, the clock on the XCK pin is used as the transmit clock.

A frame that receives 5 to 8 bits of data

Once the receiver has detected a valid start bit, it begins to accept the data. Each bit of data after the start bit will be set
Set the baud rate or XCK clock to receive until the first stop bit of a frame of data is received and the second stop bit is
Receiver ignored. Each bit of data received is sent to the receive shift register. After receiving the first stop bit, the receiver
Set the RXC bit in the receive data completion flag of the UCSRA register and transfer the complete data frame in the shift register

In the receive buffer, the CPU can obtain the received data by reading the UDR register.

A frame that receives 9-bit data

If a 9-bit data frame is set, the register UCSRB must first be read before reading the lower 8-bit data from the UDR
RXB8 bits to get the 9th bit of data. This rule also applies to the status flags FE, DOR, and PE. Read the UDR
The memory cell changes the state of the receive buffer and changes the TXB8, FE, DOR, and PE that are also stored in the buffer
Bit.

Receive end flag and interrupt handling

The USART receiver has a flag: the receive end flag RXC, which indicates whether the receive buffer has an unread
data. This bit is "1" and "0" when the receive buffer has data that is not read. If the receiver is disabled,
The receive buffer is refreshed and RXC is cleared.
Set the receive enable interrupt enable bit RXCIE after UCSRB, as long as the RXC flag is set (and the global interrupt is enabled)
The USART reception end interrupt is generated. When data is received using the interrupt mode, the data reception ends the interrupt service routine
The sequence must read the data from the UDR to clear the RXC flag, otherwise a new interrupt will be completed as soon as the interrupt handler ends
Will produce.

Receive error flag

The USART receiver has three error flags: frame error FE, data overflow DOR, and parity error PE. They are all located
UCSRA register. The error flag is stored in the receive buffer along with the data frame. All the wrong signs can not be produced
Interrupted.
The frame error flag FE indicates the state of the first stop bit of the next readable frame stored in the receive buffer. Stop bit is positive
(The value is "1"), the FE flag is "0", otherwise the FE flag is "1". This flag can be used to detect loss of synchronization
The interrupt is also available for protocol processing.
The data overflow flag DOR indicates that data loss is caused by the receive buffer. When the receive buffer is full, receive the shift
The data already exists in the register, and if a new start bit is detected at this time, a data overflow is generated. The DOR flag is set to that
Indicating that one or more data frames have been lost between the most recent read UDR and the next read UDR. When the data frame is successful
The DOR flag is cleared after shifting from the shift register to the receive buffer.
The parity error flag PE indicates that the next frame of data in the receive buffer has a parity error at reception. If no parity is enabled
Check, PE is cleared.

Parity checker

Setting the parity mode bit UPM1 will start the parity checker. The calibration mode (even parity or odd parity) is determined by UPM0
set. After the parity is enabled, the validator will calculate the parity of the input data and compare the result to the parity bit of the data frame.
The check result will be stored in the receive buffer along with the data and stop bits. The CPU checks the received frame by reading the PE bit
Whether there are parity errors. If the next data read from the receive buffer has a parity error, and the parity
Yes, the UPE is set and is always valid until the receive buffer UDR is read.

Disable the receiver

It is forbidden for the receiver to function immediately compared to the transmitter. The data being received will be lost. Disable receiver (RXEN clear)

, The receiver will no longer occupy the RxD pin and the receive buffer will be refreshed.

Asynchronous data reception

The USART has a clock recovery unit and a data recovery unit to handle asynchronous data reception. The clock recovery logic is used to synchronize from
The RxD pin inputs the asynchronous serial data and the internal baud rate clock. Data recovery logic is used to collect data and pass low
The filter filters the input of each bit of data, thereby improving the receiver's anti-jamming performance. Asynchronous reception by the scope of work
Depending on the accuracy of the internal baud rate clock, the rate of the frame input, and the number of data bits contained in one frame.

Asynchronous working range

The operating range of the receiver depends on the degree of mismatch between the received data rate and the internal baud rate. If the sender to
Too fast or too slow bit rate to transmit data, or the baud rate generated within the receiver does not have the same frequency, then the receiver
It can not be synchronized with the start bit. To ensure that the receiver does not miss the sampling of the next frame start bit, the data input rate and the internal
The baud rate of the receiver can not be too large, and the ratio between them is used to describe the baud rate error range. The following two tables are divided
Do not give the maximum permissible baud rate error range in normal mode and double speed mode.

The maximum receiver baud rate error range in normal mode

Data bit + parity bit length and	Maximum error range (%)	Recommended error range (%)

5	+ 6.7 / -6.8	± 3.0
6	+ 5.8 / -5.9	± 2.5
7	+ 5.1 / -5.2	± 2.0
8	+ 4.6 / -4.5	± 3.0
9	+ 4.1 / -4.2	± 1.5
10	+ 3.8 / -3.8	± 1.5

Maximum Receiver Baud Rate Error Range in Double Speed Mode

Data bit + parity bit length and	Maximum error range (%)	Recommended error range (%)
5	+ 5.7 / -5.9	± 2.5
6	+ 4.9 / -5.1	± 2.0
7	+ 4.4 / -4.5	± 1.5
8	+ 3.9 / -4.0	± 1.5
9	+ 3.5 / -3.6	± 1.0
10	+ 3.2 / -3.3	± 1.0

As can be seen from the table, the baud rate in normal mode allows for a greater range of variation. The recommended baud rate error range is above
Assume that the receiver and the transmitter have the same contribution to the maximum total error. Generates the receiver baud rate error
There may be two reasons. First, the stability of the receiver system clock is related to the operating voltage and temperature. Using crystal to produce
System clocks generally do not have this problem, but when using the internal oscillator, the system clock may be biased. The second reason
The baud rate generator does not necessarily get the desired baud rate by dividing the system clock. At this time can be adjusted
The value of UBRR makes the error as low as acceptable.

Baud rate setting and introduction error

For standard crystal and resonator frequencies, the baud rate for actual communication in asynchronous mode can be calculated by the baud rate calculation formula
The error between it and the commonly used baud rate can be calculated using the following formula:

Error [%] = (Baud $_{real}$ / Baud - 1) * 100%

Among them, Baud is the commonly used communication baud rate, Baud $_{real}$ is calculated by calculating the baud rate, into the baud rate meter
Calculate the relationship between the baud rate error and the system clock f $_{sys}$ and the baud rate register UBRR value as follows:
Normal mode:

Error [%] = (f $_{sys}$ / (16 * (UBRR + 1)) / Baud - 1) * 100%

Speed mode:

Error [%] = (f $_{sys}$ / (8 * (UBRR + 1)) / Baud - 1) * 100%

The baud rate error UBRR is obtained when the clock error on both sides of the communication is not considered, ie the system clock f $_{sys}$ is the standard clock
The relationship between the values. The following table shows the baud rate error for different UBRR values under the 16MHz system clock.

16MHz system clock to set the UBRR value generated by the error

Baud rate (bps)	f $_{sys}$ = 16.000MHz			
	Normal mode (U2X = 0)		Speed mode (U2X = 1)	
	UBRR	error	UBRR	error
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4K	68	0.6%	138	-0.1%
19.2K	51	0.2%	103	0.2%
28.8K	34	-0.8%	68	0.6%
38.4K	25	2.1%	34	-0.8%
57.6K	16	0.2%	51	0.2%
76.8K	12	0.2%	25	0.2%

115.2K	8	-3.5%	16	2.1%
230.4K	3	8.5%	8	-3.5%
250K	3	0%	7	0%
0.5M	1	0%	3	0%
1M	0	0%	1	0%

Multiprocessor communication mode

The Multiprocessor Communication Mode (MPCM) bits of the UCSRA are set to enable the data frame received by the USART receiver filter. Frames with no address information will be ignored and will not be stored in the receive buffer. In a multiprocessor system, each The processor communicates over the same serial bus, which effectively reduces the number of data frames that require CPU processing the amount. The MPCM bit settings do not affect the transmitter's work, but in multiprocessor communication systems, it's used Different.

If the receiver receives a data frame length of 5 to 8 bits, the first stop bit is used to indicate that the current frame contains Is data or address information. If the length of the data frame received by the receiver is 9 bits, then it is determined by the 9th bit that the data Or address information. If the frame type flag is "1", then this is the address frame, otherwise it is a data frame.

In multiprocessor communication mode, multiple slave processors are allowed to receive data from a host processor. First by decoding the address Frame to determine which address is addressed from the processor. The addressed slave processor will normally receive subsequent data while others The slave will ignore these data frames until the next address frame is received.

For a processor as a host, it can use the 9-bit data frame format and use the 9th bit of data to identify the frame formula. In this communication mode, the slave processor must also operate in a 9-bit data frame format.
The following is the multi-processor communication mode for data exchange steps:

1. All slave processors are operating in multiprocessor communication mode (set MPCM);
2. The host processor sends an address frame, and all slave processors receive this frame. From the RXC bit of the processor UCSRA register Normal set;
3. Each slave processor reads the contents of the UDR register and decodes the address frame to determine if it is selected. If selected , The MPCM bit in the UCSRA register is cleared and is not selected. Wait for MPCM to be "1" and wait for the next The arrival of an address frame;
4. The addressed slave processor receives all the data frames until a new address frame is received. Not addressed from the office The processor ignores these data frames;
5. After receiving the last data frame from the processor, the MPCM bit is set and the next address frame is set arrival. And then repeat from the second step.

The frame format using 5 to 8 bits of data is possible, but it is impractical because the receiver must use n and n + 1 frames Format to switch between. Since the receiver and transmitter use the same character length setting, this setting allows full duplex operation It becomes very difficult. If you use frame format of 5 to 8 bits of data, the transmitter should set two stop bits, where the first The stop bit is used to determine the frame type.

Register definition

UCSRA - USART Control and Status Register **A**

UCSRA - **USART** Control and Status Register **A**

Address: 0xC0 Default: 0x20

Bit	7	6	5	4	3	2	1	0
Name	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPME
R / W	R	R / W	R	R	R	R	R / W	R / W

Initial	0	0	1	0	0	0	0	0

Bit	Name	description
7	RXC	Receive end flag. When the value of RXC is "1", it indicates that there is unread data in the receive buffer. When the value of RXC is "0" , Indicating that there is no unread data in the receive buffer. When the receiver is disabled, the receive buffer is refreshed and guided Caused RXC to be cleared. When the receive end interrupt enable bit RXCIE is "1", the RXC can be used to generate the receive node Beam break.
6	TXC	Send the end flag.

5	UDRE	TXC is set when the data in the transmit shift register is sent and the transmit buffer is empty. Execute the sending end TXC is automatically cleared when it is set, or it can be cleared by writing "1" to TXC. When the transmit end interrupt is enabled When TXCIE is "1", TXC can be used to generate a transmit end interrupt. Data register empty flag. When UDRE is "1", it indicates that the USART send data buffer is empty and can write data. When UDRE is "0" indicates that the USART transmit data buffer is full and can not write data. When the data register is empty When the enable bit UDRIE is "1", the UDRE can be used to generate a data register empty interrupt.
4	FE	Frame error flag. When FE is "1", it indicates that the data received by the receive data buffer has a frame error, that is, the first stop bit is "0". When FE is "0", it indicates that there is no frame error in the data received by the receive data buffer, ie the first The stop bit is "1". FE is set to be valid until the UDR is read. When writing to UCSRA, FE This one to write "0".
3	DOR	Data overflow flag. When the receive buffer is full (contains two data), there is data in the receive shift register, and if it is detected at this time A new start bit, a data overflow occurs, DOR is set, has been valid until the UDR is read. On UCSRA To write, DOR this bit to write "0".
2	PE	Parity error flag. When the parity is enabled (UPM1 is "1"), and the received data frame in the receive buffer has a parity Error, PE is set, has been valid until UDR is read. When writing to UCSRA, this bit is required for PE Write "0".
1	U2X	Speed transmit enable bit. When U2X is "1", the transmission rate of the asynchronous communication mode is doubled. When U2X is "0", asynchronous communication The transfer rate of the mode is the normal rate. This bit is valid only in asynchronous operation mode and is cleared when the synchronous operation mode is used.
0	MPCM	multiprocessor communication mode enable bit. Setting the MPCM bit will start the multiprocessor communication mode. After the MPCM is set, the USART receiver receives it Those input frames that do not contain address information will be ignored. The transmitter is not affected by the MPCM settings.

UCSRB - USART Control and Status Register **B**

UCSRB **- USART** Control and Status Register **B**

Address: 0xC1 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	RXCIE	Receive end interrupt enable bit. Set RXC interrupt and disable RXC interrupt after clearing. When RXCIE is "1", the global interrupt is enabled, UCSRA When the RXC of the register is "1", the USART receive end interrupt can be generated.
6	TXCIE	The transmit end interrupt enable bit.

Enable TXC interrupt after setting, disable TXC interrupt after clearing. When TXCIE is "1", the global interrupt is enabled, UCSRA

When the TXC of the register is "1", the USART transmit end interrupt can be generated.

Data register empty interrupt enable bit.

5 UDRIE Set to enable UDRE interrupt after clearing, disable UDRE interrupt after clearing. When UDRIE is "1", global interrupt is enabled,

The USART data register can be generated when the UDRE of the UCSRA register is "1".

Receive enable bit.

4 RXEN Set the USART receiver after setting. The general IO function of the RxD pin is replaced by the USART reception. Prohibited to receive

The device will refresh the receive buffer and invalidate the FE, DOR and PE flags.

Send enable bit.

3 TXEN Set the USART transmitter after setting. The general IO function of the TxD pin is replaced by the USART send. TXEN clear

After zero, it will only be possible to actually disable the USART transmission until all data has been sent.

Character length control bit 2.

2 UCSZ2

UCSZ2 is combined with UCSZ1: 0 of the UCSRC register to set the number of data bits contained in the data frame.

Receive data bit 8.

1 RXB8 When the data frame length is 9 bits, RXB8 is the most significant bit of the received data. Reads the lower 8 bits of data contained in the UDR

Before you read RXB8 first.

Send data bit 8.

0 TXB8 When the data frame length is 9 bits, TXB8 is the most significant bit of the transmitted data. Write the lower 8 bits of data contained in the UDR

Before the first write TXB8.

UCSRC - USART Control and Status Register C

UCSRC - USART Control and Status Register C

Address: 0xC2 Default: 0x06

Bit	7	6	5	4	3	2	1	0
Name	UMSEL1	UMSEL0	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	1	1	0

Bit	Name	description	
		USART mode selection bit.	
		UMSEL selects synchronous or asynchronous operation mode.	
		UMSEL	mode
7: 6	UMSEL1: 0	0	USART Asynchronous operation mode
		1	USART Synchronous mode of operation
		2	SPI slave operating mode
		3	SPI master operating mode
		Parity mode selection bit.	
		High UPM1 Select to enable or disable parity, low order UPM0 Select odd parity or even parity.	
		UPM1: 0	mode
5: 4	UPM1: 0	0	Disable parity
		1	Keep it
		2	Enable even parity
		3	Enable odd parity

3	USBS	Stop bit select bit. Select the number of bits for the stop bit. USBS: Stop bit number 0: 1 1: 2
2: 1	UCSZ1: 0	Data frame character length selection bit. UCSZ1: 0 is combined with UCSZ2 of the UCSRB register to set the number of data bits contained in the data frame. UCSZ2: 0: Data frame length 0: 5 digits 1: 6 digits 2: 7 digits 3: 8 bits 4: Keep it 5: Keep it 6: Keep it 7: 9 digits
0	UCPOL	Clock polarity select bits. In the USART synchronous operating mode, UCPOL sets the output data change and the sampling of the input data And the relationship between the synchronous clock XCK. Use asynchronous operation mode with UCPOL regardless of this bit Cleared UCPOL: Send data changes: Receive data sampling 0: XCK rising edge: The falling edge of XCK 1: The falling edge of XCK: XCK rising edge

UBRRL - USART baud rate register low byte

***UBRRL* - USART** baud rate register low byte

Address: 0xC4 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	UBRR7	UBRR6	UBRR5	UBRR4	UBRR3	UBRR2	UBRR1	UBRR0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	UBRR [7: 0]	The low byte portion of the USART baud rate register. USART baud rate register contains UBRRL and UBRRH two parts, together used to set the communication Baud rate.

UBRRH - USART baud rate register high byte

***UBRRH* - USART** baud rate register high byte

Address: 0xC5 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	UBRR11	UBRR10	UBRR9	UBRR8
R / W	-	-	-	-	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description

7: 4	-	Keep it.
3: 0	UBRR [11: 8]	The high byte portion of the USART baud rate register. USART baud rate register contains UBRRL and UBRRH two parts, together used to set up through The baud rate of the letter. UBRR = {UBRR [11: 8], UBRRL}

Operating mode	Baud rate calculation formula
Asynchronous normal mode	BAUD = f_{sys} / (16 * (UBRR + 1))
Asynchronous speed mode	BAUD = f_{sys} / (8 * (UBRR + 1))
Synchronous host mode	BAUD = f_{sys} / (2 * (UBRR + 1))

UDR - USART data register

UDR - **USART** data register

Address: 0xC6 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	UDR7	UDR6	UDR5	UDR4	UDR3	UDR2	UDR1	UDR0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	UDR	USART sends and receives data. The USART transmit data buffer and the receive data buffer share the USART data register UDR. Write the data Into the UDR that is written to send data buffer, read data from the UDR to read the receive data buffer. In the 5 to 8-bit data frame mode, the unused bits 9 are ignored by the transmitter and the receiver sets them Is 0. The transmit buffer can only be written when the UDRE flag of the UCSRA register is "1", otherwise The operation of the transmitter will go wrong. When the transmit shift register is empty, the transmitter will send the data in the buffer Is loaded into the transmit shift register, and then the data is serially output from the TxD pin. The receive buffer contains a two-stage FIFO. Once the receive buffer is read, the FIFO changes its state.

USART0 - SPI operating mode

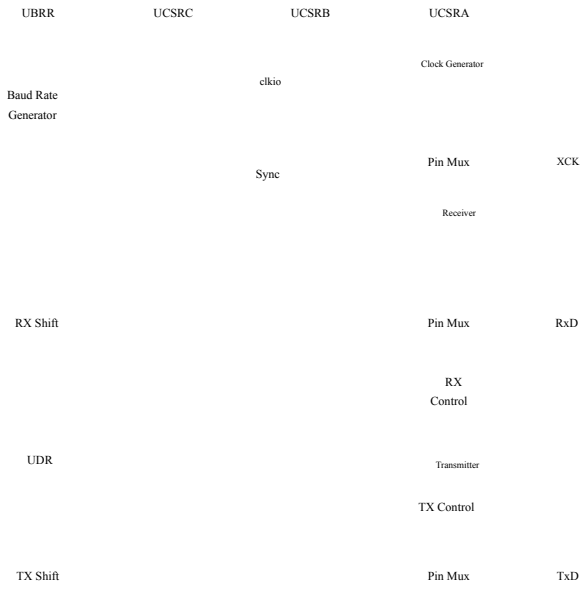
- Full-duplex operation, three-wire synchronous data transmission
- Host or slave operation
- Supports all four operating modes (modes 0, 1, 2 and 3)
- Low or high first transmission (configurable data transfer order)
- Queue operation (double buffer)

High resolution baud rate generator

Summary

When the UCREL1 bit of the USCRC is set to "1", the SPI operating mode is enabled and is represented by USPI. This SPI module is
Three-wire SPI operating mode, compared with the four-wire SPI mode, the lack of slave select line, the other three lines are consistent. USPI accounts
Use the USART resources, including sending and receiving shift registers and buffers, and baud rate generators. Parity production
Raw and check logic, data and clock recovery logic are invalid. The address of the control and status registers is the same, but send it

The function of the register bit will change as the SPI mode of operation needs.



USART in SPI structure

Clock generation

When the SPI is in host mode, it is necessary to provide a clock for communication, multiplexing the baud rate generator of the USART to generate this A clock. The clock is output from the XCK pin, so the XCK pin's data direction register (DDR_XCK) must be set Is "1".

The clock frequency is determined by the following formula:

$BAUD = f_{sys} / (2 * (UBRR + 1))$

When the SPI is operating in Slave mode, the communication clock is provided by the external host, input from the XCK pin, so the XCK pin The data direction register (DDR_XCK) must be set to "0".

SPI data mode and timing

SPI has four clock phase and polarity combination, there are control bits UCPHA and UCPOL to determine the specific control such as The following table and the following figure:

SPI operating mode

SPI mode	UCPOL	UCPHA	Starting along	End the edge
0	0	0	Rising edge sampling	Falling edge setting
1	0	1	Rising edge setting	Down edge sampling
2	1	0	Down edge sampling	Rising edge setting
3	1	1	Falling edge setting	Rising edge sampling

SPI operating mode icon

Frame format

A serial frame of the SPI can start with the least significant bit or the most significant bit, ending with the most significant bit or the lowest bit, for a total of 8 bits of data. one
After the end of the frame, you can immediately transfer the new frame, the end of the transmission can be pulled up the data line is idle.

data transmission

SPI sets the TXEN bit in the UCSRB register to "1" to enable the transmitter. The TxD pin is occupied by the transmitter to send the serial
Output data. The receiver can not be enabled at this time.
SPI sets the RXEN bit in the UCSRB register to "1" to enable the receiver. The RxD pin is occupied by the receiver to receive the serial
Input data. The transmitter must be enabled at this time.

Both SPI transmit and receive uses XCK as the transfer clock.

The SPI is first initialized before communication is performed. The initialization process usually includes the setting of the baud rate, the frame data bit
The setting of the transmission sequence, and the reception of the receiver or transmitter as needed. For interrupt-driven SPI operations, the initialization is done
The global interrupt flag is cleared and all interrupts of the SPI are disabled.
When performing a re-initialization such as changing the baud rate or frame structure, it is necessary to ensure that no data is transmitted. TXC flag can be
Used to detect whether the transmitter has completed all transmissions, the RXC flag can be used to detect whether there are still numbers in the receive buffer
It is not read. If the TXC flag is used for this purpose, before each data is sent (before writing the UDR register)
The TXC flag must be cleared.

After initializing the SPI, write data to the UDR register to start the data transfer. Since the transmitter controls the transmission clock,
Send and receive data are doing so. When sending a shift register ready to send a new frame of data, the transmitter is on
The data written to the UDR register is moved from the transmit buffer to the transmit shift register and sent out. to ensure that
The input buffer and the transmit data are synchronized, and the UDR register must be read once every byte of data has been transmitted. When made
When data is spilled, the most recently received data will be lost, not the earliest received data.

Send flag and interrupt

The SPI transmitter has two flags: the SPI data register empty flag UDRE and the transmission end flag TXC, both flag bits
Can generate an interrupt.

The data register empty flag UDRE is used to indicate whether the transmit buffer can write a new data. The bit is sending slow
When the punch is empty, it is set to "1" and "0" when it is full. When the UDRE bit is "1", the CPU can write to the data register UDR
Into the new data, and vice versa can not.

When the data register empty interrupt enable bit UDRIE in the UCSRB register is "1", as long as the UDRE is set (and
Global interrupt enable), the SPI data register will generate an empty interrupt request. Writing to the register UDR will be cleared
UDRE. When transmitting data in an interrupt mode, a new number must be written in the data register empty interrupt service routine
To the UDR to clear the UDRE, or to disable the data register from being interrupted. Otherwise, once the interrupt service routine ends,
A new interrupt will be generated again.

When the entire data frame is shifted out of the transmit shift register and there is no new data in the transmit register, the transmit end flag

TXC will be set. Set the transmit enable bit to TXCIE (and global interrupt enable) on UCSRB to "1"
When the TXC flag is set, the SPI transmit end interrupt will be executed. Once the interrupt service routine is entered, the TXC is marked
The bit is automatically cleared, the CPU can also write "1" to the bit to clear.

Disable the transmitter

When TXEN is cleared, only after all the data has been sent to send the transmitter can be really prohibited, that is, send the shift
There is no data to be transferred in the register and transmit buffer registers. After the transmitter is disabled, the TxD pin resumes its generic
IO function.

Receive end flag and interrupt

The SPI receiver has a flag: the receive end flag RXC, which indicates whether the receive buffer has an unread number

according to. This bit is "1" and "0" when the receive buffer has data that is not read. If the receiver is banned
The receive buffer will be refreshed and the RXC will be cleared.
Set the receive enable interrupt enable bit RXCIE after UCSRB, as long as the RXC flag is set (and the global interrupt is enabled)
The SPI receive end interrupt is generated. When data is received using the interrupt mode, the data reception end interrupt service routine must be completed
Read the data from the UDR to clear the RXC flag, otherwise a new interrupt will be generated as soon as the interrupt handler is completed
Health

Disable the receiver

It is forbidden for the receiver to function immediately compared to the transmitter. The data being received will be lost. Disable the receiver (RXEN clear
Zero), the receiver will no longer occupy the RxD pin and the receive buffer will be refreshed.

Register definition

USART register list

register	address	Defaults	description
UCSRA	0xC0	0x20	USPI Control and Status Register
UCSRB	0xC1	0x00	USPI Control and Status Register B
UCSRC	0xC2	0x06	USPI Control and Status Register C
UBRRL	0xC4	0x0	USPI baud rate register low byte
UBRRH	0xC5	0x0	USPI baud rate register high byte
UDR	0xC6	0x0	USPI data register

UCSRA - USPI control and status register **A**

In UCSRA - **in USP** Control and Status Register **A**

Address: 0xC0 Default: 0x20

Bit	7	6	5	4	3	2	1	0
Name	RXC	TXC	UDRE	-	-	-	-	-
R / W	R	R / W	R	-	-	-	-	-
Initial	0	0	1	0	0	0	0	0

Bit Name Description

7	RXC	Receive end flag. When the value of RXC is "1", it indicates that there is unread data in the receive buffer. When the value of RXC is "0" Indicating that there is no unread data in the receive buffer. When the receiver is disabled, the receive buffer is refreshed, resulting in RXC Was cleared. When the receive end interrupt enable bit RXCIE is "1", the RXC can be used to generate a receive end interrupt.
		Send the end flag. TXC is set when the data in the transmit shift register is sent and the transmit buffer is empty. Execute the sending end

6	TXC	TXC is automatically cleared when it is set, or it can be cleared by writing "1" to TXC. When sending an end interrupt enable bit When TXCIE is "1", TXC can be used to generate a transmit end interrupt.
5	UDRE	Data register empty flag. When UDRE is "1", it indicates that the USPI send data buffer is empty and can write data. When UDRE is

		"0" indicates that the USPI sends the data buffer to full and can not write data. When the data register is empty When the UDRIE bit is "1", the UDRE can be used to generate a data register empty interrupt.
4: 0	-	Retained under USPI.

UCSRB - USPI Control and Status Register B

UCSRB - USPI Control and Status Register B

Address: 0xC1 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	RXCIE	TXCIE	UDRIE	RXEN	TXEN	-	-	-
R / W	R / W	R / W	R / W	R / W	R / W	-	-	-
Initial	0	0	0	0	0	0	0	0

Bit	Name	Description
7	RXCIE	Receive end interrupt enable bit. Set RXC interrupt and disable RXC interrupt after clearing. When RXCIE is "1", global interrupt is enabled, The USPI receive end interrupt can be generated when the RXC of the UCSRA register is "1".
6	TXCIE	The transmit end interrupt enable bit. Enable TXC interrupt after setting, disable TXC interrupt after clearing. When TXCIE is "1", global interrupt is enabled, When the TXC of the UCSRA register is "1", the USPI transmission end interrupt can be generated.
5	UDRIE	Data register empty interrupt enable bit. Set to enable UDRE interrupt after clearing, disable UDRE interrupt after clearing. When UDRIE is "1", global interrupt is enabled, The UDPI of the UCSRA register is "1" to generate a USPI data register empty interrupt.
4	RXEN	Receive enable bit. Set the USPI receiver after setting. The general IO function of the RxD pin is replaced by the USPI reception. Disable the receiver The receive buffer will be refreshed.
3	TXEN	Send enable bit. Set the USPI transmitter after setting. The general IO function of the TxD pin is replaced by the USPI transmission. TXEN cleared , It is only after all data has been sent to complete the USART can not really be sent.
2: 0	-	Retained under USPI.

UCSRC - USART Control and Status Register C

UCSRC - USART Control and Status Register C

Address: 0xC2 Default: 0x86

Bit	7	6	5	4	3	2	1	0
Name	UMSEL1	UMSEL0	-	-	-	DORD	UCPHA	UCPOL
R / W	R / W	R / W	-	-	-	R / W	R / W	R / W
Initial	0	0	0	0	0	1	1	0

Bit	Name	description
7: 6	UMSEL1: 0	USART mode selection bit.

UMSEL selects synchronous or asynchronous operation mode.

UMSEL	mode
0	USART Asynchronous operation mode
1	USART Synchronous mode of operation
2	SPI slave operating mode
3	SPI master operating mode

5: 3 - Retained under USPI.

2 DORD Data transfer order selection bit.

DORD	Data order
0	High first transmission
1	Low first transmission

1 UCPHA Clock phase selection.

UCPHA Selects data sampling at the beginning or end edge.

UCPHA	Sampling time
0	Starting along
1	End the edge

0 UCPOL Clock polarity selection.

UCPOL selects data changes and samples occur on rising or falling edges.

UCPOL	Send data changes	Sampling of received data
0	XCK rising edge	The falling edge of XCK
1	The falling edge of XCK	XCK rising edge

UBRRL - USPI baud rate register low byte

UBRRL - **USPI** baud rate register low byte

Address: 0xC4 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	UBRR7	UBRR6	UBRR5	UBRR4	UBRR3	UBRR2	UBRR1	UBRR0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	UBRR [7: 0]	USPI The low byte portion of the baud rate register. The USPI baud rate register contains both UBRRL and UBRRH Part, combined to set the baud rate for communication.

UBRRH - USPI baud rate register high byte

The UBRRH - **in USP** Baud Rate Register High Byte

Address: 0xC5 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	UBRR11	UBRR10	UBRR9	UBRR8
R / W	-	-	-	-	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 4	-	Retained under USPI.
3: 0	UBRR [11: 8]	The high byte portion of the USPI baud rate register. USPI baud rate register contains UBRRL and UBRRH two parts, together used to set the communication Baud rate. UBRR = {UBRR [11: 8], UBRRL}

Operating mode	Baud rate calculation formula
Slave mode	The baud rate is determined by the external host
Host mode	BAUD = f_{sys} / (2 * (UBRR + 1))

UDR - USPI data register

***UDR* - USPI** data register

Address: 0xC6 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	UDR7	UDR6	UDR5	UDR4	UDR3	UDR2	UDR1	UDR0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	UDR	USPI sends and receives data. The USPI sends the data buffer and the receive data buffer to share the USPI data register UDR. Write the data UDR is written to send data buffer, read data from the UDR read the receive data buffer. In the 5 to 8-bit data frame mode, the unused 9th bit is ignored by the transmitter, and the receiver sets them Set to 0. Only when the UDRE flag of the UCSRA register is "1" can the write buffer be written, no The operation of the transmitter will go wrong. When the transmit shift register is empty, the transmitter will send the buffer in the buffer The data is loaded into the transmit shift register, and the data is serially output from the TxD pin. The receive buffer contains a two-stage FIFO. Once the receive buffer is read, the FIFO changes its shape state.

TWI - Dual Line Serial Bus *(I2C)*

- Simple and powerful and flexible communication interface, only 2 lines
- Supports host and slave operation
- The device can operate in transmitter mode or receiver mode
- There are 128 slaves in the 7-bit address space
- Multi-host arbitration is supported
- Up to 400Kbps data transfer rate
- Fully programmable slave address and public address
- You can wake up when the address matches in sleep mode

TWI bus introduction

Two-wire serial interface TWI is well suited for typical processor applications. The TWI protocol allows the system designer to use only two bi-directional
The transmission line can interconnect 128 different devices together. The two lines are the clock SCL and the data SDA. External hardware
Only need to connect two pull-up resistors on each line. All devices connected to the bus have their own addresses. TWI protocol solution
Determine the problem of bus arbitration.

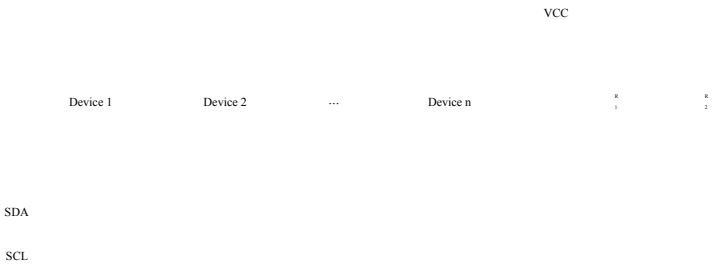
TWI terms

The terms defined below will appear frequently in this section.

the term	description
Host	Start and stop the transmission of the device. The host is also responsible for generating the SCL clock.
Slave machine	The device that is addressed by the host
Transmitter	Place the data on the bus
receiver	A device that receives data from the bus

Electrical connections

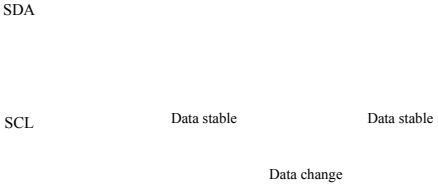
As shown in the following figure, the two lines of the TWI interface are connected to the positive supply via the pull-up resistor. All TWI-compatible devices are driven by the bus
Moving are open-drain or open-collector, so that the interface to achieve the operation of the line and function. When the TWI device is output
When "0", the TWI bus will be low. When all TWI devices are tri-stated, the bus allows pull-up resistors
Raise the voltage. To ensure that all bus operations, all devices connected to the TWI bus must be powered on.



TWI bus interconnection diagram

Data transmission and frame structure

Each bit of data transfer on the TWI bus is synchronized with the clock. When the clock line is high, the level on the data line must be guaranteed
Hold steady, unless it is to generate a start or stop state.



TWI data validity graph

Start and stop status

TWI's transmission is initiated and stopped by the host. The host sends a START status on the bus for transmission of data and sends a STOP
State to stop data transfer. Between the START and STOP states, the bus is considered busy and does not allow other hosts

Trying to take control of the bus. There is a special case that only allows the occurrence of a new between the START and STOP states
Of the START state, which is called the REPEATED START state, applies to the current host without giving up the bus control

Start a new transmission. After the REPEATED START until the next STOP, the bus is still considered busy.

This is consistent with START, so in this document, if there is no special instructions, are used START to express START and

REPEATED START. As shown in the following figure, the START and STOP conditions change the level of the SDA line when the SCL line is high
status.

START, REPEATED START and STOP status diagrams

Address packet format

All address packets transmitted on the TWI bus are 9-bit data length, consisting of 7-bit addresses, 1-bit READ / WRITE control bits, and

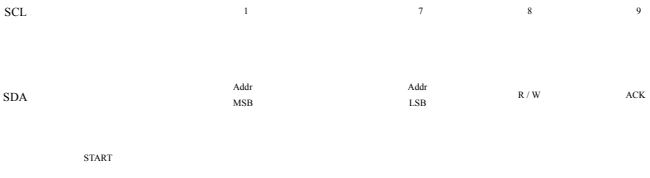
1 bit response bit. When the READ / WRITE bit is "1", a read operation is performed; when the READ / WRITE bit is "0"
Perform a write operation. After the slave is addressed, it must acknowledge in the 9th SCL (ACK) cycle by pulling down the SDA line. If
If the slave is busy or there are other reasons that can not respond to the host, the SDA line should remain high for the ACK cycle. Then the host can be made
The STOP status or REPEATED START status is restarted.

The address pack includes a slave address and a read or write control bit, denoted by SLA + R or SLA + W, respectively.

The MSB bit of the address byte first occurs. Except that the reserved address "00000000" is reserved for broadcast calls as well as all shapes
The address of the "1111xxxx" format needs to be reserved for future use. Other slave addresses can be freely assigned by the designer.

When a broadcast call occurs, all slaves should respond at the ACK cycle by pulling down the SDA line. When the host needs to be made
The broadcast function can be used when sending the same information to multiple slaves. When the broadcast call address plus the WRITE bit is sent to the total
After the line, all the slaves that need to respond to the broadcast call will pull down the SDA line during the ACK cycle. All of which responded widely
The incoming call of the slave will receive the following packet. It should be noted that sending the broadcast call address plus the READ bit is not
Meaningful, because if several slaves simultaneously send different data will bring bus conflict.

The address package format is shown below:

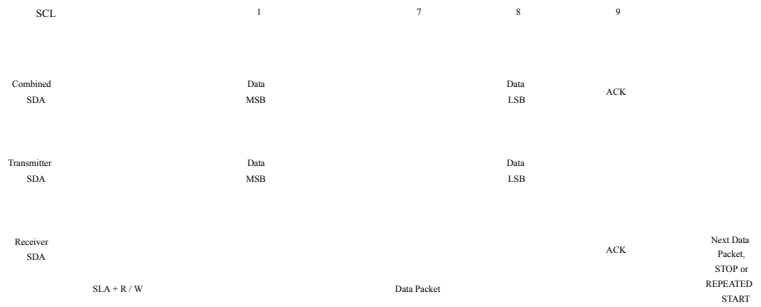


TWI address package format

Packet format

All data packets transmitted on the TWI bus are 9-bit data length consisting of 1 data byte and 1 bit acknowledge bit. in
During data transmission, the host is responsible for generating the transfer clock SCL and START and STOP status, the transmitter sends the word to be transmitted
Section data, the receiver generates a receive response. The acknowledgment signal ACK is the receiver in the 9th SCL (ACK) cycle by pulling down
SDA line to produce. If the receiver holds the SDA line high during the ACK cycle, an unacknowledged signal NACK is issued.
When the receiver has received the last byte, or for some reason can no longer receive any data, it should be in the collection
To the last byte by sending a NACK to inform the sender. The MSB bits of the data byte are transmitted first.

The packet format is shown in the following figure:



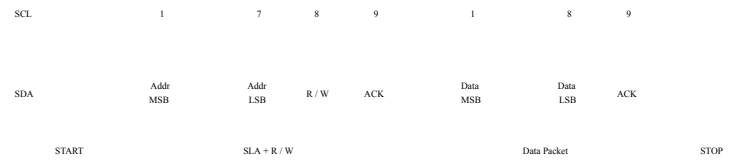
TWI packet format

Combined address and packet transmission

One transmission consists essentially of 1 START, 1 SLA + R / W, 1 or more packets, and 1 STOP. only

START and STOP empty messages are illegal. You can use the line and function of the SCL line to achieve the handshake between the master and the slave.
The slave can extend the SCL level by pulling down the SCL line. When the host set the clock speed is much faster than the slave,
Or the slave takes extra time to process the data, this feature is very useful. The slave extends the low period of SCL and
Does not affect the high period of SCL, it is still determined by the host. It can be seen that the slave can be changed by changing the SCL
Duty cycle to reduce the TWI data transfer speed.

The following figure shows a typical data transfer. Note that multiple bytes can be transferred between SLA + R / W and STOP, depending on
Application software implementation protocol.



Typical TWI transmission

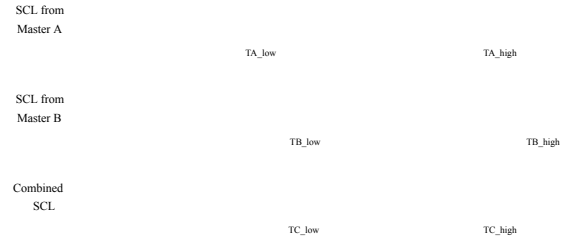
Multi - host system and its arbitration and synchronization

The TWI protocol allows multiple hosts on the bus and uses special measures to ensure that even if two or more hosts start simultaneously
The transmission can also be handled like an ordinary transmission. Multi-host system will be two problems:

1. Implementation of the algorithm allows only one host in a host to complete the transmission. When other hosts find that they have lost their options
   To stop their transmission. The process of selection is called arbitration. When the competition in the host found its arbitration failed,
   Should immediately switch to the slave mode to detect whether it is subject to bus control by the host. In fact, more host with the same
   When the transmission should not be detected by the slave, that is not allowed to destroy the data being transmitted on the bus.
2. Different hosts may use different SCL frequencies. In order to ensure the consistency of transmission, you must design a synchronous host string
   Line clock program. This will simplify the arbitration process.

The lines and functions of the bus are used to solve the above problems. The serial clocks of all the hosts will line up together to produce a group
Clock, its high time is equal to the shortest of all host clocks, the low level is equal to all the host clock
A long one. All hosts are listening to SCL, and when the combined SCL clock goes high or low, they can be effectively started separately
Calculate the respective SCL high and low level overflow periods.

Multi-host SCL clock synchronization mechanism as shown below:



Page 164

Multi - host SCL clock synchronization timing diagram

After the output data, all hosts continue to listen on the SDA line to implement the arbitration. If the value read from SDA is lost with the host
The value does not match, the host that is lost arbitration. Note that the host outputs a high level of SDA while the other host
Output low-level SDA will lose arbitration. The host that lost the arbitration should immediately switch to the slave mode and detect if it is
Be addressed. The host that lost the arbitration must set the SDA line high, but it can also be produced before the current data or address pack ends
Raw clock signal. Arbitration will continue until the system has only one host, which may take multiple bits. If multiple hosts
For the same slave addressing, the arbitration will continue to the packet.

Arbitration between two hosts

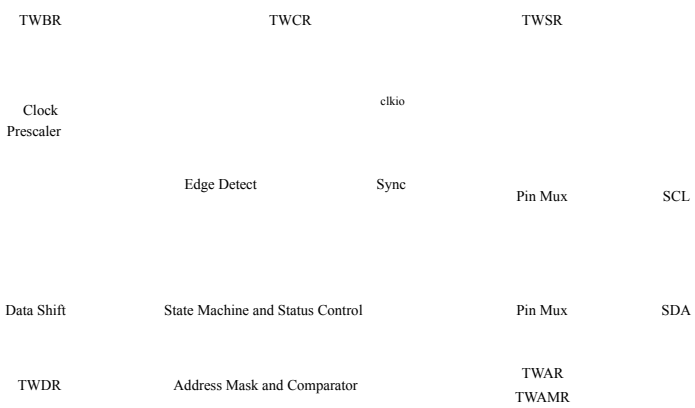
Note that arbitration is not permitted in the following cases:

- A REPEATED START state with a data bit between;
- A STOP state and a data bit between;
- Between a REPEATED START state and a STOP status;

Application software must consider the above situation, to ensure that these illegal arbitration will not appear. This means that in a multi-master system,
All data transfers must consist of the same SLA + R / W and data packets. In other words, all transmissions must contain phases
With the same number of packets, otherwise the arbitration results can not be defined.

Summary of TWI modules

The structure of the TWI module is shown in the following figure.



TWI Block structure diagram

TWI module mainly includes bit rate generator, bus interface unit, address comparator and control unit. See details below
Detailed description.

Bit rate generator unit

The bit rate generator unit mainly controls the SCL clock period in master mode. The SCL clock period consists of the TWI bit rate register
TWBR and the prescaler control bits in the TWI status register TWSR. Slave operation is not subject to bit rate or prescaler
Set the effect, but to ensure that the working clock of the slave is at least 16 times the SCL frequency. Note that the slave may extend SCL
Of the low-level period, thereby reducing the TWI bus average clock frequency. The SCL clock frequency is generated by the following calculation formula:

$$f_{scl} = f_{sys} / (16 + 2 * TWBR * 4^{TWPS})$$

Where TWBR is the value of the TWI bit rate register and TWPS is the prescaler control bit in the TWI status register.

Bus interface unit

The bus interface unit includes data and address shift register TWDR, START / STOP controller and arbitration decision hardware circuit.

TWDR contains the address or data byte to be sent, or the address or data byte that has been received. In addition to the 8-bit TWDR,
The bus interface unit also includes an ACK / NACK register that is sent or received. This ACK / NACK register can not be directly addressed
Accessed by software. When receiving data, it can be set or cleared by the TWI control register TWCR. When sending data
, The received ACK / NACK value is reflected by the TWS value in the TWI status register TWSR.

The START / STOP controller is responsible for generating and detecting START, REPEATED START and STOP statuses. When the MCU is in some
Sleep mode, START / STOP controller can still detect START and STOP status, when the host on the TWI bus search
The MCU will wake up from Sleep mode.

If TWI initiates a data transfer in host mode, the arbitration detection circuit will continuously monitor the bus to determine if it still owns
Bus control. When the TWI module loses bus control, the control unit will perform the correct action and generate the appropriate

Status code to inform the MCU.

Address matching unit

The address matching unit is used to check whether the received address byte matches the 7-bit address in the TWI address register. when
The TWI broadcast call identification enable bit (TWGCE) in the TWAR register is set and the address received from the bus is
Broadcast address comparison. Once the address matches successfully, the control unit will perform the correct action. The TWI module can respond or not respond
The addressing of the host depends on the setting of the TWCR register. Even in sleep mode, the address matching unit can be compared
Address, if the host on the bus address, the MCU wake from the sleep mode.

control unit

The control unit is responsible for monitoring the bus and generating a corresponding response according to the settings of the TWCR. Occurs when the TWI bus needs to be applied
When the software participates in the event, the TWI interrupt flag bit TWINT will be set. In the next one clock cycle, TWI-like
The status register TWSR will be updated to indicate the status code for the event. When TWINT is set, TWSR contains the exact
status information. At other times, TWSR is a special status code, indicating that there is no exact status information. Once TWINT
The flag is set and the SCL line remains low, pausing the TWI transmission on the bus, allowing the application to process the event.

The TWINT flag is set in the following cases:

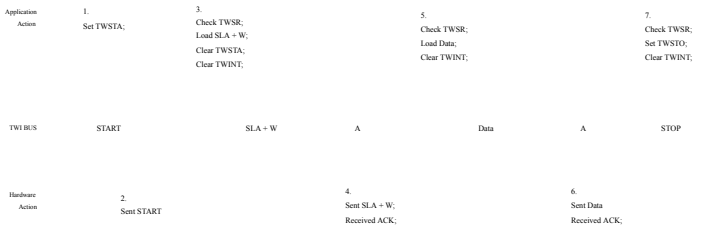
- TWI sends START / REPEATED START status
- TWI transmits SLA + R / W
- TWI sends an address byte after
- TWI bus after arbitration failed
- TWI is addressed by host (slave address match or broadcast mode)
- When it is addressed as a slave, it receives a STOP or REPEATED START
- When a bus error caused by an illegal START or STOP status occurs

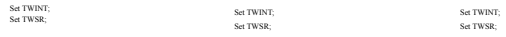
TWI use

The TWI interface is byte-oriented and interrupt-based. All bus events, such as receiving a byte or sending a START
Signal, etc., will produce a TWI interrupt. Since TWI is based on the interrupt, so in the TWI byte transmission process,
Application software can be free to carry out other operations. TWECT register in TWCR interrupt enable bit TWIE and global interrupt enable
Can be used to control whether a TWI interrupt is generated when the TWINT flag is set. If the TWIE bit is cleared, the application software
The TWINT flag must be used to detect the action on the TWI bus.

When the TWINT flag is set, it indicates that the TWI interface completes the current operation and waits for the response of the application software. At this
In the case where the TWI status register TWSR contains a status code that reflects the current bus status. Application software can pass
Set the TWCR and TWDR registers to determine how the TWI interface will work at the next TWI bus cycle.

The following figure shows an example of how the application connects to the TWI interface. In this example, the host expects to send a byte of data to
Slave machine. The description here is very simple, the next chapter will be more detailed display.





TWI typical transmission process diagram

The TWI transmission process shown in the figure is:

1. The first step in TWI transmission is to send START. The TWI hardware is sent by writing a specific value to the TWCR register
   START signal. The values written will be described in detail later. It is very important to set TWINT in the written value,
   Writing a "1" to the TWINT bit clears this bit. TWI register TWINT set during TWI does not start any operation
   For. Once the software clears the TWINT bit, the TWI module immediately initiates the transmission of the START signal.
2. When the START status is transmitted, the TWINT flag of the TWCR is set, the TWSR is updated to the new status code,
   Indicates that the START signal was sent successfully.
3. The application looks at the value of TWSR and determines that the START status has been successfully sent. If TWSR is displayed as a different value,
   The application can perform special operations, such as calling error handlers. When the status code is determined to be consistent with the expected,
   The program loads the value of SLA + W into the TWDR register. The TWDR register can be used both in address and data.
   The software then writes a specific value to the TWCR register indicating the value of the SLA + W in the TWI hardware to send TWDR. write
   The incoming value will be described in detail later. Set TWINT in the written value to clear the TWINT flag. TWCR
   The TWINT of the register is set during TWI without initiating any operation. Once the software clears the TWINT bit, the TWI module
   Immediately start the delivery of the address packet.
4. When the address packet is sent, the TWINT flag of TWCR will be set and TWSR will be updated to the new status code.
   The address packet is sent successfully. The status code also reflects whether the slave responds to the address packet.
5. The application looks at the value of TWSR, determines that the address packet has been successfully sent, and the received ACK is the expected value. If TWSR
   Displayed as other values, the application can perform some special operations, such as calling an error handler. When determining the state
   When the code is consistent with the expected, the program loads the value of Data into the TWDR register. Then the software goes to the TWCR register
   Writes a specific value that indicates the value of Data in TWIT hardware to send TWDR. The values written will be described in detail later.
   Set TWINT in the written value to clear the TWINT flag. TWINT register of TWCR register is set
   TWI does not start any operation. Once the software clears the TWINT bit, the TWI module immediately initiates the transfer of the packet.
6. When the packet is sent, the TWINT flag of TWCR is set, TWSR is updated to the new status code,
   The packet is successfully sent. The status code also reflects whether the slave responds to the packet.
7. The application looks at the value of TWSR to determine that the packet has been successfully sent and the received ACK is the expected value. If TWSR
   Displayed as other values, the application can perform some special operations, such as calling an error handler. When determining the state
   When the code is consistent with the expected, the software writes a specific value to the TWCR register indicating that the TWI hardware sends a STOP signal. write
   The incoming value will be described in detail later. Set TWINT in the written value to clear the TWINT flag. TWCR
   The TWINT of the register is set during TWI without initiating any operation. Once the software clears the TWINT bit, the TWI module
   The STOP signal is transmitted immediately. It should be noted that TWINT will not be done after the STOP signal has been transmitted
   Position.

Although the example is simpler, it contains all the rules in the TWI data transfer process. Summarized as follows:

- The TWINT flag is set when TWI completes an operation and waits for feedback from the application. The SCL clock line is always held

   Pull down until TWINT is cleared;
- When the TWINT flag is set, the user must update all TWI register values for the next TWI bus cycle
   The value. For example, the TWDR register must load the value to be sent by the next bus cycle.
- After updating all the registers while completing other necessary operations, the application writes the TWCR register. in
   When TWCR is written, the TWINT bit must be set to clear the TWINT flag. After TWINT is cleared, TWI is on
   The operation set by TWCR is executed.

Transmission mode

TWI can work in the following four main modes: Host Transmitter (MT), Host Receiver (MR), Slave Transmitter
(ST) and slave receiver (SR). Multiple modes can be used under the same application. For example, TWI can use the MT mode
TWI EEPROM writes data and reads data from EEPROM in MR mode. If there are other hosts on the system, there are
It is also possible to send data to TWI, which will use SR mode. This is the application software to decide which mode to use.

These patterns will be described in detail below. In each mode of data transmission, will be combined with pictures to describe the possible shape
Code. These pictures contain the following abbreviations:

S: Start state

Rs: REPEATED START state

R: Read operation flag (SDA is high)

W: write operation flag (SDA is low)

A: acknowledge bit (SDA is low)

NA: No acknowledge bit (SDA is high)

Data: 8-bit data bytes

P: STOP status

SLA: Slave Address

The circle in the picture is used to indicate that the TWINT flag is set. The number in the circle indicates the status code in the TWSR register. The prescaler control bit is masked as "0". In these places, the application must perform the appropriate action to continue or complete the TWI transmission. TWI transfers are pending until the TWINT flag is cleared.

When the TWINT flag is set, the status code in TWSR is used to determine the appropriate software operation. Each form is given in each form The code required for the software operation and subsequent serial transmission details. Note that the prescaler control bits in TWSR are masked Is "0".

Host send mode

In the master transmit mode, TWI will send a certain number of data bytes to the slave receiver. In order to enter the host mode, Send a START signal. The next address packet format determines whether TWI is in the host transmitter mode or the host receiver mode. If you send SLA + W, enter the host send mode. If SLA + R is sent, it enters the host receive mode. This The status codes mentioned in one section assume that the prescaler control bit is "0".

Write the START signal by writing the following values to the TWCR register:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	1	0	x	1	0	x

TWEN bit must be set to "1" to enable TWI interface, TWSTA set to "1" to send START signal, TWINT set to "1"

To clear the TWINT flag. The TWI module detects the bus status and sends the START signal as soon as the bus is idle. When sending After START, the hardware sets the TWINT flag and updates the TWSR status code to 0x08.

In order to enter the host send mode, you must send SLA + W. This can be done by doing the following. Write to the TWDR register first Into SLA + W, and then write "1" to the TWINT bit to clear the TWINT flag to continue transmission, that is, write to the TWCR register The following values are used to send SLA + W:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	0	x	1	0	x

When SLA + W is transmitted and the acknowledge signal is received, TWINT is set and the status code of TWSR is updated. may The status code is 0x18, 0x20 or 0x38. A suitable response to each status code is described in detail in the status code table.

When SLA + W is sent successfully, it can start sending packets. This can be done by writing data to the TWDR register.

TWDR can only be written when the TWINT flag is high. Otherwise, the access is ignored while writing the conflicting flag bit TWWC Will be set. After updating the TWDR, write "1" to the TWINT bit to clear the TWINT flag to continue transmission. Ie to TWCR The register writes the following values to send the data:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	0	x	1	0	x

When the packet is sent and the acknowledge signal is received, TWINT is set and the status code of the TWSR is updated. possible The status code is 0x28 or 0x30. A suitable response to each status code is described in detail in the status code table.

When the data is sent successfully, you can continue to send data packets. This process is repeated until the last byte is sent. The host generates a STOP signal or a REPEATED START signal before the entire transmission ends.

The STOP signal is issued by writing the following values to the TWCR register:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	1	x	1	0	x

Write the REPEATED START signal by writing the following values to the TWCR register:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	1	0	x	1	0	x

After sending REPEATED START (status code 0x10), the TWI interface can access the same slave again, or access New slave without sending a STOP signal. REPEATED START causes the host to be able to lose control of the bus Switching between different slaves, master transmitter and host receiver modes.

The status code in the host transmit mode and the corresponding operation are shown in the following table:

Host status code table for sending mode

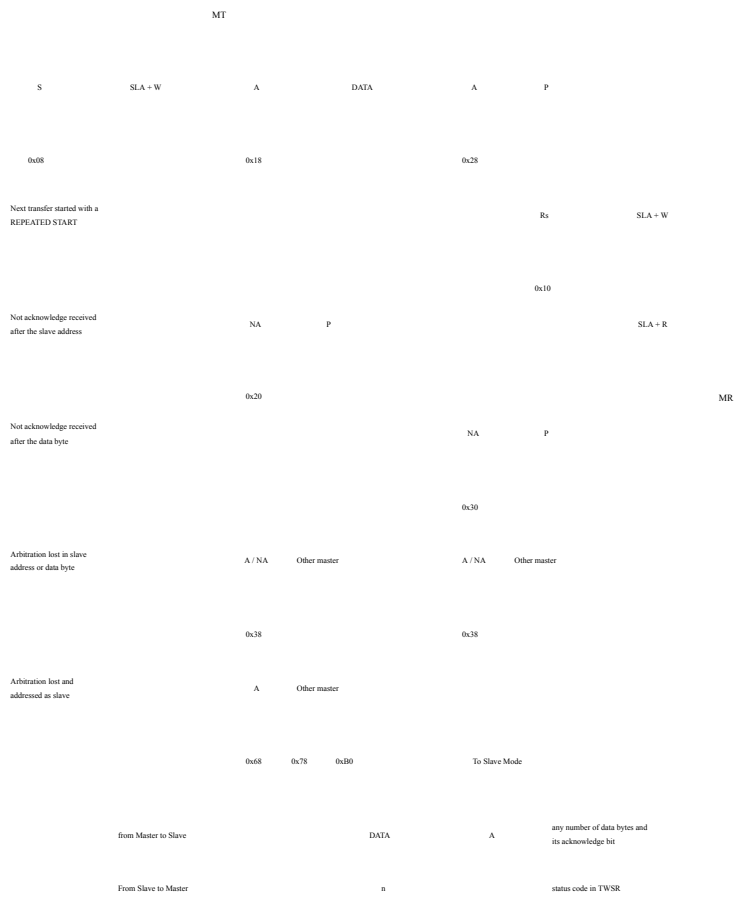
status code	Bus and hardware status	Application software response					The next step in the hardware
		Read / write	Operation of **TWCR**				
		TWDR	**STA**	**STO**	**TWINT**	**TWEA**	
0x08	START has been sent to load	0	0	1	x	Will send SLA + W;	
		SLA + W					ACK or NACK will be received
0x10	REPEATED	load	0	0	1	x	Will send SLA + W;
	START has been sent	SLA + W					ACK or NACK will be received

		load	0	0	1	x	Will send SLA + R;
		SLA + R					Will receive ACK or NACK;
							Will switch to MR mode
0x18	SLA + W has been issued	Number of loads	0	0	1	x	Will send data;
	give away;	according to					ACK or NACK will be received
	Received ACK	No operation	1	0	1	x	Will send REPEATED START
		No operation	0	1	1	x	Will send STOP;
							The TWSTO flag is reset
		No operation	1	1	1	x	Will send STOP;
							The TWSTO flag will be reset;
							Will send START
0x20	SLA + W has been issued	Number of loads	0	0	1	x	Will send data;
	give away;	according to					ACK or NACK will be received
	Received NACK	No operation	1	0	1	x	Will send REPEATED START
		No operation	0	1	1	x	Will send STOP;
							The TWSTO flag is reset
		No operation	1	1	1	x	Will send STOP;
							The TWSTO flag will be reset;
							Will send START
0x28	Data bytes already	Number of loads	0	0	1	x	Will send data;
	Send;	according to					ACK or NACK will be received
	ACK	No operation	1	0	1	x	Will send REPEATED START
		No operation	0	1	1	x	Will send STOP;
							The TWSTO flag is reset
		No operation	1	1	1	x	Will send STOP;
							The TWSTO flag will be reset;
							Will send START
0x30	Data bytes already	Number of loads	0	0	1	x	Will send data;
	Send;	according to					ACK or NACK will be received
	NACK	No operation	1	0	1	x	Will send REPEATED START
		No operation	0	1	1	x	Will send STOP;
							The TWSTO flag is reset

		No operation	1	1	1	x	Will send STOP; The TWSTO flag will be reset; Will send START
0x38	SLA + W or number Failed by arbitration	No operation	0	0	1	x	Will release the bus; Will enter the unaddressed slave mode
		No operation	1	0	1	x	Will send START when idle

The format and status of the host send mode are as follows:

Page 171



The format and status of the host send mode

Host receive mode

In host receive mode, TWI receives a certain number of data bytes from the slave transmitter. In order to enter the host mode, Send a START signal. The next address packet format determines whether TWI is in the host transmitter mode or the host receiver mode. If you send SLA + W, enter the host send mode. If SLA + R is sent, it enters the host receive mode. This The status codes mentioned in one section assume that the prescaler control bit is "0".

Write the START signal by writing the following values to the TWCR register:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE

1	x	1	0	x	1	0	x

TWEN bit must be set to "1" to enable TWI interface, TWSTA set to "1" to send START signal, TWINT set to "1"
To clear the TWINT flag. The TWI module detects the bus status and sends the START signal as soon as the bus is idle. When sending

Page 172

After START, the hardware sets the TWINT flag and updates the TWSR status code to 0x08.

In order to enter the host receive mode, you must send SLA + R. This can be done by doing the following. Write to the TWDR register first
Into SLA + R, and then write "1" to the TWINT bit to clear the TWINT flag to continue transmission, ie write to the TWCR register
The following values are used to send SLA + R:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	0	x	1	0	x

When the SLA + R is transmitted and the acknowledge signal is received, TWINT is set and the status code of the TWSR is updated. possible
The status code is 0x38, 0x40, or 0x48. A suitable response to each status code is described in detail in the status code table.

When SLA + R is sent successfully, it can start receiving packets. Clear the TWINT flag by writing "1" to the TWINT bit
Continue to receive. The following values are written to the TWCR register to initiate reception:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	0	x	1	0	x

When the packet is received and the acknowledge signal is transmitted, TWINT is set and the status code of the TWSR is updated. possible
The status code is 0x50 or 0x58. A suitable response to each status code is described in detail in the status code table.

When the data reception is successful, you can continue to receive data packets. This process is repeated until the last byte is received.
After the host receives the last byte, it must send a NACK acknowledge signal to the slave transmitter. The host generates a STOP signal
Or the REPEATED START signal is completed by the entire reception.

The STOP signal is issued by writing the following values to the TWCR register:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	0	1	x	1	0	x

Write the REPEATED START signal by writing the following values to the TWCR register:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
1	x	1	0	x	1	0	x

After sending REPEATED START (status code 0x10), the TWI interface can access the same host again, or access
The new host without sending a STOP signal. REPEATED START causes the host to be able to lose control of the bus
Switching between different slaves, master transmitter and host receiver modes.

The status code in the host receive mode and the corresponding operation are shown in the following table:

Host receive mode status code table

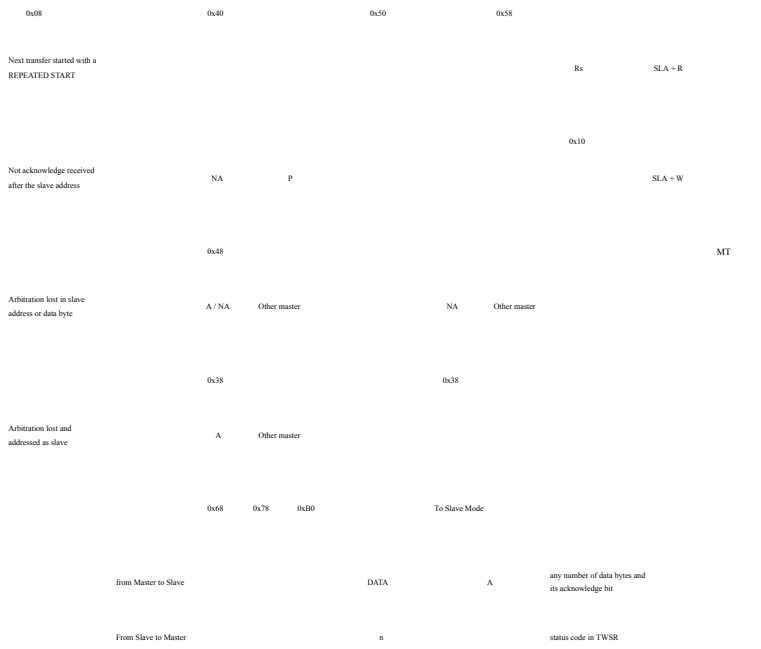
status code	Bus and hard State of the item	Application software response: Read / write TWDR	STA	Operation of TWCR: STO	TWINT	TWEA	The next step in the hardware
0x08	START has been issued give away	load SLA + R	0	0	1	x	Will send SLA + R; ACK or NACK will be received
0x10	REPEATED START has been issued give away	load SLA + R load SLA + W	0	0	1	x	Will send SLA + R; ACK or NACK will be received Will send SLA + W; Will receive ACK or NACK;

							Will switch to MT mode
0x38	SLA + R or number	No operation	0	0	1	x	Will release the bus;
	According to arbitration,						Will enter the unaddressed slave mode
	defeat	No operation	1	0	1	x	Will send START when idle
0x40	SLA + R has been issued	No operation	0	0	1	0	Will receive data;
	give away;						Will send NACK
	Received ACK	No operation	0	0	1	1	Will receive data;
							ACK will be sent
0x48	SLA + R has been issued	No operation	1	0	1	x	Will send REPEATED START
	give away;	No operation	0	1	1	x	Will send STOP;
	received						The TWSTO flag is reset
	NACK	No operation	1	1	1	x	Will send STOP;
							The TWSTO flag will be reset;
							Will send START
0x50	Data bytes	Read the number	0	0	1	0	Will receive data;
	Received;	according to					Will send NACK
	ACK has been sent	Read the number	0	0	1	1	Will receive data;
		according to					ACK will be sent
0x58	Data bytes	Read the number	1	0	1	x	Will send REPEATED START
	Received;	according to					
	NACK has been issued	Read the number	0	1	1	x	Will send STOP;
	give away	according to					The TWSTO flag is reset
		Read the number	1	1	1	x	Will send STOP;
		according to					The TWSTO flag will be reset;
							Will send START

The format and status of the host receive mode are as follows:

MR

S SLA + R A DATA A DATA NA P



The format and status of the host receive mode

Slave receive mode

In the slave receive mode, a certain number of data bytes can be received from the host transmitter. The status code mentioned in this chapter
It is assumed that the prescaler control bit is "0".

To start the slave receive mode, set the TWAR and TWCR registers.
TWAR needs to be set as follows:

TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Device slave address							

The upper 7 bits of TWAR are the slave addresses that the TWI interface responds to when the host is addressed. If LSB is set, TWI responds to broadcast call
Called address (0x00), otherwise ignore the general call address.
TWCR should be set as follows:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
0	1	0	0	0	1	0	x

TWEN must be set to enable the TWI interface, TWEA must be set to enable the host to address (slave address or broadcast call) to
Return to the confirmation message ACK. TWSTA and TWSTO must be cleared.

After the TWAR and TWCR are initialized, the TWI interface starts waiting until its slave address (or broadcast address) is found
site. The TWI enters the slave receive mode when the data direction bit of the slave address is "0" (indicating a write operation). when
When the data direction bit is "1" (indicating read operation), TWI enters the slave transmission mode. Receive their own slave address and write
After the operation flag is set, the TWINT flag is set and the valid status code is also updated to TWSR. All status codes are appropriate
The response will be described in detail in the status code table. It should be noted that the host mode after the TWI arbitration can also be failed
To enter the slave receive mode (see status codes 0x68 and 0x78).

If the TWEA bit is reset during transmission, TWI will return NACK (high) to SDA after receiving a byte
on-line. This can be used to indicate that the slave can not receive more data. When the TWEA bit is "0", TWI will not respond
The address of the slave. However, TWI will still monitor the bus, once TWEA is set, you can restore the address recognition and response.
That is, you can use TWEA to temporarily isolate the TWI interface from the bus.

The clock of the TWI interface can be turned off in other sleep modes except the idle mode. If the slave can receive the mode,
The interface will continue to respond to the slave address or broadcast address using the bus clock. Address match will wake up the MCU. During awakening,
The TWI interface will remain SCL low until the TWINT flag is cleared. When the TWI interface clock returns to normal
Receive more data.

The status code of the slave receive mode is shown in the following table:

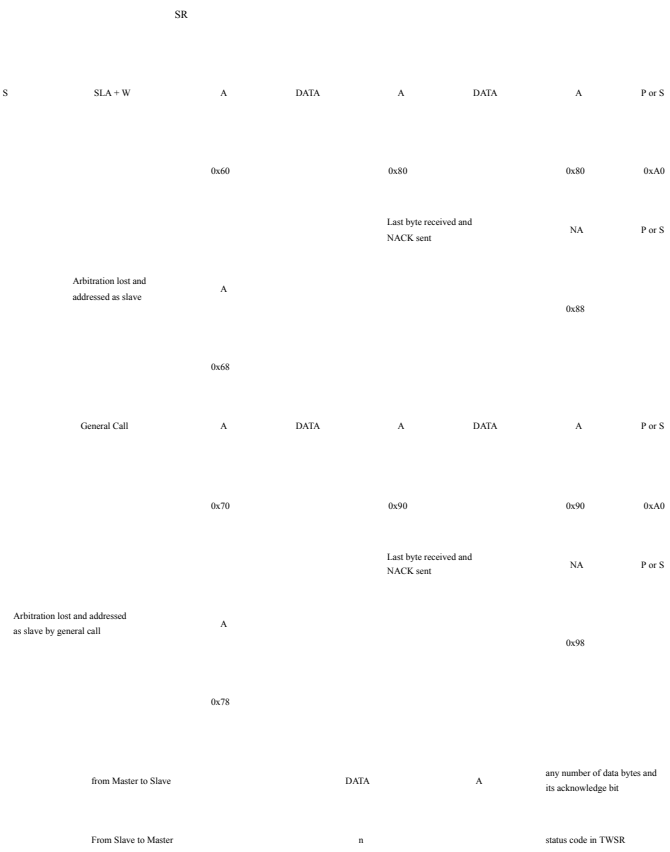
Slave receive mode status code table

		Application software response					
status code	Bus and hardware status	Read / write	Operation of **TWCR**				The next step in the hardware
		TWDR	**STA**	**STO**	**TWINT**	**TWEA**	
0x60	SLA + W has been received;	No operation	x	0	1	0	Will receive data;
	ACK has been sent						Will send NACK
		No operation	x	0	1	1	Will receive data;
							ACK will be sent
0x68	Sending SLA + R / W	No operation	x	0	1	0	Will receive data;
	Failing						Will send NACK
	SLA + W has been received;	No operation	x	0	1	1	Will receive data;
	ACK has been sent						ACK will be sent
0x70	The broadcast address has been received;	No operation	x	0	1	0	Will receive data;
	ACK has been sent						Will send NACK
		No operation	x	0	1	1	Will receive data;
							ACK will be sent
0x78	Sending SLA + R / W	No operation	x	0	1	0	Will receive data;
	Failing						Will send NACK
	SLA + W has been received;	No operation	x	0	1	1	Will receive data;
	ACK has been sent						ACK will be sent
0x80	Own data has been received;	Read the number	x	0	1	0	Will receive data;
	ACK has been sent	according to					Will send NACK
		Read the number	x	0	1	1	Will receive data;
		according to					ACK will be sent
0x88	Own data has been received;	Read the number	0	0	1	0	Will switch to unreserved from

	NACK has been sent	according to					Machine mode
							Will not respond to the slave address
							And broadcast
		Read the number	0	0	1	1	Will switch to unreserved from
		according to					Machine mode
							Will respond to the slave address;
							TWGCE = 1 will respond
							broadcast
		Read the number	1	0	1	0	Will switch to unreserved from
		according to					Machine mode
							Will not respond to the slave address
							And broadcast;
							Bus will be sent when idle
							START
		Read the number	1	0	1	1	Will switch to unreserved from
		according to					Machine mode
							Will respond to the slave address;
							TWGCE = 1 will respond
							broadcast;
							Bus will be sent when idle

						START
0x90	Broadcast data received; ACK has been sent	Read the number according to	x	0	1	0	Will receive data; Will send NACK
		Read the number according to	x	0	1	1	Will receive data; ACK will be sent
0x98	Broadcast data received; NACK has been sent	Read the number according to	0	0	1	0	Will switch to unreserved from Machine mode Will not respond to the slave address And broadcast
		Read the number according to	0	0	1	1	Will switch to unreserved from Machine mode Will respond to the slave address; TWGCE = 1 will respond broadcast
		Read the number according to	1	0	1	0	Will switch to unreserved from Machine mode Will not respond to the slave address And broadcast; Bus will be sent when idle START
		Read the number according to	1	0	1	1	Will switch to unreserved from Machine mode Will respond to the slave address; TWGCE = 1 will respond broadcast; Bus will be sent when idle START
0xA0	When the machine is working STOP or REPEATED START	No operation	0	0	1	0	Will switch to unreserved from Machine mode Will not respond to the slave address And broadcast
		No operation	0	0	1	1	Will switch to unreserved from Machine mode Will respond to the slave address; TWGCE = 1 will respond broadcast
		No operation	1	0	1	0	Will switch to unreserved from Machine mode Will not respond to the slave address And broadcast; Bus will be sent when idle START
		No operation	1	0	1	1	Will switch to unreserved from Machine mode Will respond to the slave address; TWGCE = 1 will respond broadcast; Bus will be sent when idle START

The format and status of the slave receive mode are as follows:



Slave receive mode format and status diagram

Slave send mode

In the slave transmit mode, a certain number of data bytes can be sent to the host receiver. The status code mentioned in this chapter

It is assumed that the prescaler control bit is "0".

To start the slave receive mode, set the TWAR and TWCR registers.

TWAR needs to be set as follows:

TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
Device slave address							

The upper 7 bits of TWAR are the slave addresses that the TWI interface responds to when the host is addressed. If LSB is set, TWI responds to broadcast call Called address (0x00), otherwise ignore the general call address.

TWCR should be set as follows:

TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
0	1	0	0	0	1	0	x

TWEN must be set to enable the TWI interface, TWEA must be set to enable the host to address (slave address or broadcast call) to Return to the confirmation message ACK. TWSTA and TWSTO must be cleared.

After the TWAR and TWCR are initialized, the TWI interface starts waiting until its slave address (or broadcast address) is found site. The TWI enters the slave receive mode when the data direction bit of the slave address is "0" (indicating a write operation). when When the data direction bit is "1" (indicating read operation), TWI enters the slave transmission mode. Receive their own slave address and read After the operation flag is set, the TWINT flag is set and the valid status code is also updated to TWSR. All status codes are appropriate The response will be described in detail in the status code table. It should be noted that the host mode after the TWI arbitration can also be failed To enter the slave transmit mode (see status code 0xB0).

If the TWEA bit is reset during transmission, TWI will switch to unaddressed slave mode after sending the last byte. The status code in the TWSR register will be updated after the host receiver gives NACK or ACK for the last byte of transmission. For 0xC0 or 0xC8. If the host receiver continues to transmit the operation, the slave transmitter will not respond, the host will receive the full "1" data (ie 0xFF). When the slave sends the last byte of data (TWEA is cleared) and the NACK is expected Response, and the host wants to receive more data to send ACK as a response, TWSR will be updated to 0xC8.

When the TWEA bit is "0", TWI does not respond to its own slave address. But TWI will still listen to the bus once TWEA Is set, you can restore the address recognition and response. That is, you can use TWEA to temporarily switch the TWI interface from the bus Isolated out.

The clock of the TWI interface can be turned off in other sleep modes except the idle mode. If the slave can receive the mode, The interface will continue to respond to the slave address or broadcast address using the bus clock. Address match will wake up the MCU. During awakening, The TWI interface will remain SCL low until the TWINT flag is cleared. When the TWI interface clock returns to normal Receive more data.

The status code of the slave transmit mode is shown in the following table:

Slave status code table for sending mode

status code	Bus and hard State of the item	Application software response Read / write TWDR	Operation of TWCR STA	STO	TWINT	TWEA	The next step in the hardware
0xA8	SLA + R has been connected Close ACK has been sent	Load data	x	0	1	0	Will send the last data; It is desirable to receive NACK
		Load data	x	0	1	1	Will send data; ACK will be received
0xB0	hair give away SLA + R / W Failure of arbitration; SLA + R has been connected Close ACK has been sent	Load data	x	0	1	0	Will send the last data; It is desirable to receive NACK
		Load data	x	0	1	1	Will send data; ACK will be received
0xB8	Data has been issued Send; receive	Load data	x	0	1	0	Will send the last data; It is desirable to receive NACK
		Load data	x	0	1	1	Will send data;

Page 180

							ACK will be received
0xC0	Data has been issued to	No operation	0	0	1	0	Will switch to unreserved slave mode formula;
	give away; NACK is connected						Will not respond to the slave address and broadcast
	Receive	No operation	0	0	1	1	Will switch to unreserved slave mode formula; Will respond to the slave address; When TWGCE = 1, the response will be broadcast
		No operation	1	0	1	0	Will switch to unreserved slave mode formula; Will not respond to the slave address and wide broadcast; Bus is sent when the bus is idle
		No operation	1	0	1	1	Will switch to unreserved slave mode formula; Will respond to the slave address; TWGCE = 1 will respond to the broadcast; Bus is sent when the bus is idle
0xC8	the last one Data has been issued	No operation	0	0	1	0	Will switch to unreserved slave mode formula;
	give away; ACK has been received						Will not respond to the slave address and broadcast
		No operation	0	0	1	1	Will switch to unreserved slave mode formula; Will respond to the slave address; When TWGCE = 1, the response will be broadcast
		No operation	1	0	1	0	Will switch to unreserved slave mode formula; Will not respond to the slave address and wide broadcast; Bus is sent when the bus is idle
		No operation	1	0	1	1	Will switch to unreserved slave mode formula; Will respond to the slave address; TWGCE = 1 will respond to the broadcast; Bus is sent when the bus is idle

The format and status of the slave transmit mode are shown in the following figure:

Page 181

The format and status of the slave mode

Other status

There are two status codes that do not have a corresponding TWI state definition, as shown in the following table:

Other status code table

status code	Bus and hardware status	Application software response: Read / write TWDR	Operation of TWCR: STA	STO	TWINT	TWEA	The next step in the hardware
0xF8	Stateless information; TWINT = 0	No operation	Do not operate TWCR				Wait for or perform the current operation
0x00	Illegal START Or STOP The bus is wrong	No operation	0	1	1	x	Only affect the internal hardware; Send a STOP to the bus; The bus is released and cleared TWSTO bit

The status code 0xF8 indicates that there is no relevant information at this time because the TWINT flag is "0". This state may occur at TWI
The port is not involved in serial transmission or the current transmission has not yet been completed.

State 0x00 indicates that a bus error occurred during serial transmission. Bus error when illegal START or STOP occurs
It will happen. For example, there is START or STOP between address and data, address, and ACK. The bus error will be set
TWINT. In order to recover from the error, TWSTO must be set and cleared by writing "1" to clear TWINT. This will make TWI
The interface enters the unaddressed slave mode without generating a STOP, and releases SCL and SDA, and clears the TWSTO bit.
Combination mode

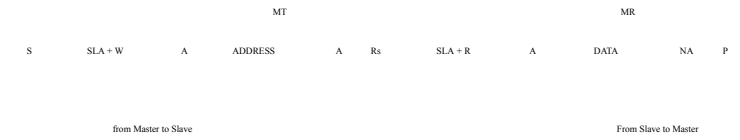
In some cases, several TWI modes must be combined to achieve the desired job. For example, from the serial EEPROM

Read the data, the typical transmission includes the following steps:
1. The transmission must be initiated;
2. You must tell the EEPROM where the data should be read;
3. must complete the read operation;
4. The transmission must end.
Note that data can be transferred from the host to the slave, and vice versa. The host tells the slave to read the location of the data, using the master
Machine sending mode. Next, read data from the slave, using the host receive mode. The direction of transmission will change. Host
Must maintain the various stages of the bus control, all the steps are uninterrupted operation. If in a multi-host system,

Between 2 and 3 another host to change the location of the data read, then break the principle of the host to read the data location
it is wrong. The direction of the data transfer is changed by sending a REPEATED START between the transfer address byte and the received data

To achieve. After sending REPEATED START, the host still has bus control.

The following figure describes the transmission process:



Combine multiple TWI modes to access the serial EEPROM diagram

Multi - host system and arbitration

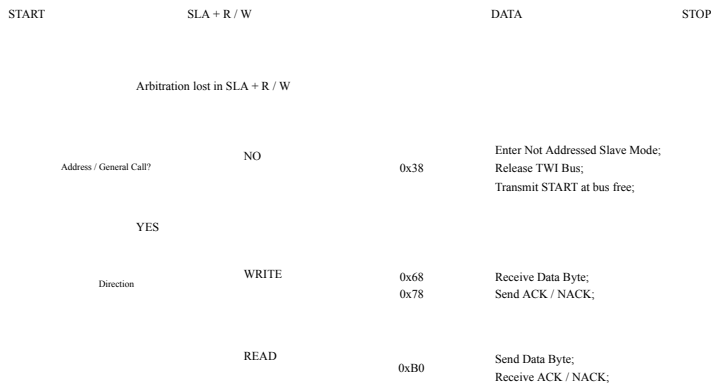
If more than one host is connected to the same TWI bus, one or more of them may start the data transfer at the same time.

The TWI protocol ensures that in this case, through an arbitration process, one of the hosts is allowed to transmit and will not be lost data. Here the two host to try to send data from the machine as an example to describe the bus arbitration process.

There are several different situations that will result in a bus arbitration process:

- Two or more hosts communicate with a slave at the same time. In this case, both the host and the slave do not know
  There is competition on the bus;
- Two or more hosts simultaneously access different data or operating directions for the same slave. In this case will be
  Arbitration occurs at the READ / WRITE bit or data bit. When there are other hosts to the SDA line to send "0", to SDA
  A host that sends "1" online will fail the arbitration. The failed host will switch to unreserved slave mode, or
  Waiting for the bus to send a new START signal, which depends on the operation of the application software.
- Two or more hosts access different slaves. In this case, bus arbitration occurs in the SLA phase. When there is
  When the host sends a "0" to the SDA line, the host that sends "1" to the SDA line will fail the arbitration. In SLA total
  A host that fails during a quorum will switch to slave mode and check if it is being addressed by the host of the bus control.
  If addressed, it will enter SR or ST mode, depending on the READ / WRITE bit following the SLA. If not
  Addressing, it will switch to unreserved slave mode, or wait for the bus to be idle when sending a new START message
  Number, depending on the operation of the application software.

The following figure describes the process of bus arbitration:



Bus arbitration process

Register definition

TWI register list

register	address	Defaults	description
TWBR	0x B8	0x00	TWI bit rate register
TWSR	0xB9	0x00	TWI status register
TWAR	0xBA	0x00	TWI address register
TWDR	0xBB	0x00	TWI data register
TWCR	0xBC	0x00	TWI control register
TWAMR	0xBD	0x00	TWI address mask register

TWBR - TWI bit rate register

TWBR - **TWI** bit rate register

Address: 0xB8 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	TWBR7	TWBR6	TWBR5	TWBR4	TWBR3	TWBR2	TWBR1	TWBR0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	TWBR [7: 0]	TWI bit rate selection control bit. TWBR is the bit rate generator division factor. The bit rate generator is a divider used in the host Mode generates an SCL clock. The bit rate is calculated as follows: $f_{scl} = f_{sys} / (16 + 2 * TWBR * 4^{TWPS})$.

TWSR - TWI status register

TWSR - **TWI** status register

Address: 0xB9 Default: 0xF8

Bit	7	6	5	4	3	2	1	0
Name	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	1	1	1	1	1	0	0	0

Bit	Name	description
7: 3	TWS [7: 3]	TWI status flag. 5-bit TWS response TWI logic and bus status. Different status values have different meanings, specific See the description of TWI operating mode. The value read from TWSR includes a 5-bit status value and a 2-bit pre-score Frequency control bit, in the detection state should be blocked pre-divide bit is "0". This is state detection independent of pre-division Frequency setting.
2	-	Keep it.
1	TWPS1	TWI prescaler control high. TWPS1 and TWPS0 together form TWPS [1: 0] to control the bit rate prescaler, and TWBR Together control the bit rate.
0	TWPS0	TWI prescaler control low. TWPS0 and TWPS1 together form TWPS [1: 0] to control the bit rate prescaler, and TWBR Together control the bit rate.

TWPS [1: 0]	Prescaler factor
0	1

1	4
2	16
3	64

TWAR - TWI address register

TWAR - TWI address register

Address: 0xBA Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	TWAR6	TWAR5	TWAR4	TWAR3	TWAR2	TWAR1	TWAR0	TWGCE
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 1	TWA [6: 0]	TWI slave address bits. TWA is the TWI slave address. When TWI is operating in slave mode, TWI will enter this address Line response. Host mode does not require this address. But in the multi-host system, also need to set the slave address to Other host access.

0	TWGCE	TWI broadcast identification enable control bit. When the TWGCE bit is set to "1", the TWI bus broadcast identification is enabled. When setting the TWGCE bit to "0", the TWI bus broadcast recognition is disabled. When TWGCE is set and the received address frame is 0x00, the TWI module responds to this bus broadcast.

TWDR - TWI data register

TWDR - TWI data register

Address: 0xBB Default: 0xFF

Bit	7	6	5	4	3	2	1	0
Name	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	1	1	1	1	1	1	1	1

Bit	Name	description
7: 0	TWD [7: 0]	TWI data register. TWD is the next byte to be transmitted on the bus, or the last byte received from the bus byte.

TWCR - TWI control register

TWCR - TWI control register

Address: 0xBC Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
R / W	R / W	R / W	R / W	R / W	R	R / W	-	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
		TWI interrupt flag. When TWI completes the current job and wants the application to intervene, the hardware sets the TWINT bit. If the whole

7	TWINT	When the bit is set and the TWIE bit is set, a TWI interrupt will be generated and the MCU will execute the TWI interrupt service routine. When the TWINT flag is set, the low level of the SCL signal will be extended. The TWINT flag can only be cleared by writing a "1" to this bit. Even if the interrupt service routine is executed, The hardware will not automatically clear the bit. Also note that clearing this bit will immediately turn on TWI. because In this case, before clearing the TWINT bit, first complete the TWAR, TWAMR, TWSR and TWDR Access to the device.
6	TWEA	TWI enables response control bits. The TWEA bit controls the generation of the acknowledge pulse. When the TWEA bit is set to "1", one of the following conditions is satisfied , A response pulse will be generated on the TWI bus: 1) Receives the slave address of the device; 2) received a call when TWGCE is set;

		3) Receive one byte of data in the host receive or slave receive mode. When the TWEA bit is set to "0", the device is temporarily disconnected from the TWI bus. After the device is set, Complex address recognition.
5	TWSTA	TWI start status control bit. The TWSTA bit needs to be set when the CPU wishes to become a host on the TWI bus. The hardware will detect the total Line is available, when the bus is idle, the bus on the start state. When the bus is not idle, TWI will wait until the detection of the stop state occurs, and then generate a starting state to declare that they want to become Host. The software must clear the TWSTA bit after sending the start status.
4	TWSTO	TWI stop status control bit. In the host mode, when the TWSTO bit is "1", TWI will generate a stop state on the bus, and then Clear the TWSTO bit. In the slave mode, setting the TWSTO bit enables the TWI to be restored from the error state come. This will not produce a stop state, only let TWI return to a defined unreserved from Mode, while releasing the SCL and SDA signal lines to the high impedance state.
3	TWWC	TWI write conflicting flag. When the TWINT flag is low, writing the TWDR register bit will be set in the TWDR register. When TWINT When the flag bit is high, writing the TWDR register will clear the TWWC flag.
2	TWEN	TWI enable control bit. The TWEN bit enables TWI operation and activates the TWI interface. When the TWEN bit is set to "1", TWI is controlled The IO pin is connected to the SCL and SDA pins. When the TWEN bit is set to "0", the TWI interface module is turned off Closed, all transmissions are terminated, including ongoing operations.
1	-	Keep it.
0	TWIE	TWI interrupt enable control bit. When the TWIE bit is set to "1" and the global interrupt is set, the TWINT flag is high Active TWI interrupt request.

TWAMR - TWI address mask register

TWAMR - **TWI** address mask register

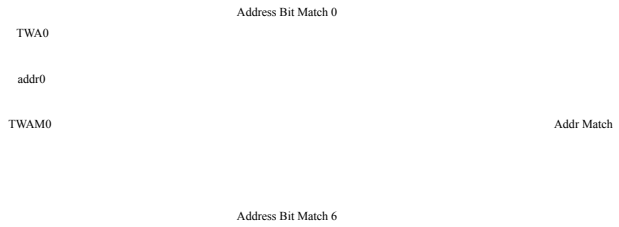
Address: 0xBD Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	TWAR6	TWAR5	TWAR4	TWAR3	TWAR2	TWAR1	TWAR0	TWGCE
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 1	TWAM [6: 0]	TWI address mask control bit. TWAM is a 7-bit TWI slave address mask control. Each bit of TWAM is used to block (disable) The corresponding address bits in TWAR. When the mask bit is set, the address match logic ignores the received address Bit and TWA corresponding bit comparison results. The following figure shows the details of the address matching logic.
0	-	Keep it.

TWI address matching logic

The following figure shows the TWI address matching logic block diagram:



TWI Address Matching Logical Structure

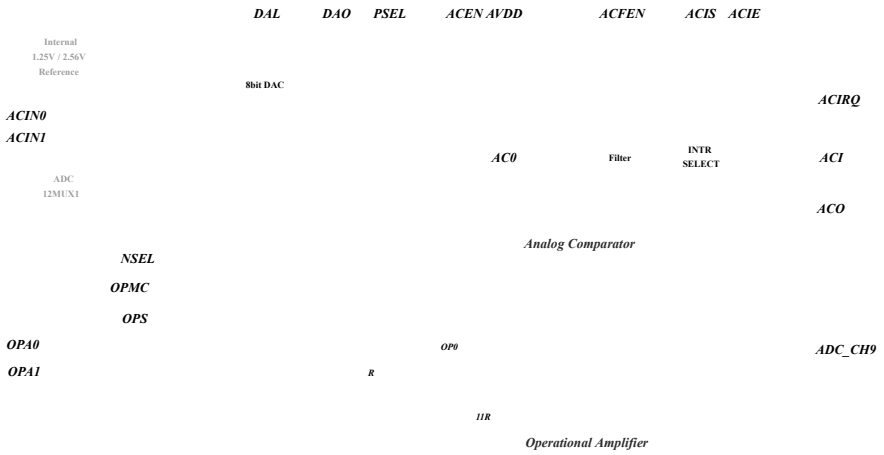
Op amp / comparator *0*

- 12mV more accurate
- Supports 2 off-chip analog inputs
- Supports multiplexing output from ADC
- Supports output from internal OPAMP op amp
- Supports internal 1.25 / 2.56V reference voltage input
- Internal integrated 8-bit DAC analog-to-digital converter
- With the pre-amplifier circuit to achieve over-current / over-voltage protection
- Programmable output filter control

Summary

The op amp / comparator module integrates an input mode configurable front stage op amp and a built-in 8-bit DAC analog ratio
Compared to the front; op amp support can be configured forward / reverse input mode, fixed forward 12 times, reverse 11 times the amplification gain,
Can be combined with analog comparator 8-bit DAC, to achieve a flexible signal detection; op amp 0 output can also be through the modulus
Converter (ADC) for more accurate processing; analog comparator on the positive value and the value of the negative comparison, when the positive pole
Is higher than the voltage on the negative pole, the output of the analog comparator ACO is set. When the level of ACO changes,
The edge of the signal can be used to trigger an interrupt. The output signal ACO can also be used to trigger the input capture and pairing of the timer counter 1
The PWM output generated by the timer is controlled. The analog comparator integrates an 8-bit precision digital-to-analog converter (DAC)
The internal reference voltage can be decomposed into 256 different reference voltage levels.

The structure of the op amp / analog comparator 0 is shown in the following figure.



Op amp / comparator 0 module structure diagram

Input of the front op amp

The OPA0 analog front end is responsible for the multiplex input of the analog input channels, the forward / reverse input mode switching, and the input signal
Do magnification. OPA0 amplified signal is divided into two, respectively, connected to the ADC input channel and analog comparator (AC)
Of the negative input. In other words, the op amp processing signal, you can directly use the comparator for fast processing, you can also

Through ADC conversion to get accurate results, which can achieve more complex functions.

OPA0 analog all part of the OPA0 controller under the control of the work; OPA0 analog part of the switch input or change Input mode, the output has a stable process, OPA0 controller is responsible for generating the controller timing, to avoid not Stable output has an effect on the post-stage circuit.

The preamplifier supports two analog input channels: OPA0 / 1. Two channels can be switched through software, can also pass The OPA controller inside the chip automatically switches from time to time. The channel switching time has an 8-bit timer individually controlled, Timer clock source can be selected for the system clock or internal RC32M 32-way (1MHz), to achieve more flexible Of the channel switching cycle to meet the needs of different applications.

When the op amp function is enabled, it is necessary to set the control bits associated with the op amp pin input of the DIDR1 register to avoid I / O numbers The functional part has an effect on the op amp's analog input channel. For details, refer to the Register Definitions section of this chapter.

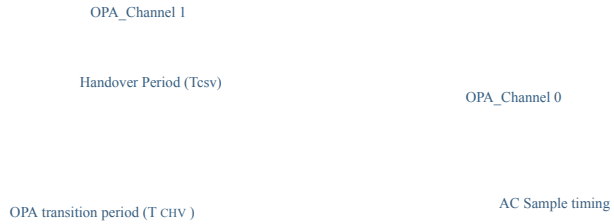
Opener channel switching mechanism

In the automatic timing channel switching mode, in order to avoid the interference generated during the channel switching process, the channel switching action and simulation Comparator output filter linkage, to achieve a smooth channel switch. Users can also adjust the associated registers to achieve Complete control of the switching process.

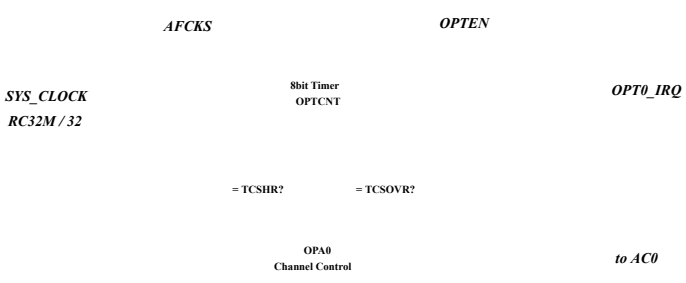
OPA0 channel switching state machine is responsible for real-time switching input channel, you can only one op amp on the basis of two The effect of the road input. OPA0 switches the channel and switches the input mode corresponding to the channel. So we can To achieve two positive input, two reverse input and all the way forward all the way to the three input combinations.

OPA0 in the process of switching the channel, the op amp output will have an unstable change, the channel switching status opportunities in the The switching channel simultaneously generates a control signal which is used to control the next stage circuit for processing the op amp output. Such as analog comparators, analog comparators can use this control information to its output to do the appropriate filtering process, so as to avoid Road switching produces instability and interference.

Channel switching timing:



As shown above, T CSV is the OPA0 channel switching cycle, OPA0 start channel switching, there is an inherent build time (T CHV), during which time the output of OPA0 may be unstable or can not immediately respond to the current state of the channel, By configuring the T CHV cycle, let the next level filter out the interference introduced by the switching channel.



OPA0 *OP0*
OPA1 *R*

OPA0 *11R*

OPA0 channel switching state machine contains an internal 8-bit counter, used to set the channel switching cycle (T CSV),
The count clock can be the default system clock, or you can select 32 of the 32MHz RC oscillator on the chip
(1 MHz). So that the user can choose the appropriate switching cycle according to the specific application requirements. For some reaction to trigger
Requires fast applications, such as overcurrent protection, you can choose a faster count clock, configure a smaller counting cycle. In addition to the timer
Used to generate T CSV , also used to generate T CHV cycles. The user can control the timing of channel switching through registers.

The channel switching timer, in addition to the timing used to generate channel switching, can also be used as a separate 8-bit timer
Use, the timer overflow after the interrupt signal. The user can switch the input channel using the software by interrupting the service. in order to
So that the software switch channel is more convenient, OPA0 controller to achieve a software switching channel dedicated control register (OP0CRA)
The user only needs to write 1 to the SCSW bit in the register to implement a channel switch. When using the software timer to switch
Channel, you can still configure the OP0CRB to achieve T CHV timing control.

The user can also use the channel switching timer as a timer that is completely independent of the OPA0 controller.
Other systems require timing tasks. The software only needs to be set by setting the OPTEN bit of the OP0CRB register
Enable timer, timer count clock selection is also a TCKCSR register AFCKS bit control, timer overflow
The preset is set by the OP0TCNT register.

Analog comparator input

Both inputs of the analog comparator support a variety of optional input sources. The positive input supports the off-chip pins ACIN0 and
The 8-bit DAC, based on an internal reference, is selected by the ACBG in the ACSR of the AC control status register
Bit to control, see the register description. The negative input supports the off-chip pin ACIN1, the output of the ADC multiplexer, and from
Op amp OPAMP output. The comparator negative input channel is selected by the ADCCSRB register from the ADC module
ACME00 / 01 bits to control. When ACIN1 is selected for configuration, the AIND1 bit of the DIDR1 register must be set at the same time,
Otherwise the input of ACIN1 will not be selected for the negative input of the analog comparator.

The following table shows the input control table for the analog comparator.

AC0 negative input control

ACME01	ACME00	DIDR1 [1]	MUX [2: 0]	AC Negative Input
0	0	0	xxx	ADC [xxx]
0	0	1	xxx	AIN1
0	1	1	xxx	ADC [xxx]
0	1	x	000	ADC0
0	1	x	001	ADC1
0	1	x	010	ADC2
0	1	x	011	ADC3
0	1	x	100	ADC4
0	1	x	101	ADC5
0	1	x	110	ADC6
0	1	x	111	ADC7
1	x	x	x	OPA0

Comparator output filter

Analog comparator real-time response to the input signal changes, when the input signal on the interference, the same analog comparator
The output will appear similar to the interference or instantaneous transition. This interference may result in erroneous operation. So in the simulation
Comparator output, in series with a filter circuit; filter circuit is divided into two parts, first with the op amp OPA0 with

Use the output hold circuit, the OPA0 output analog input comparator input, if the open OPA0
The automatic channel switching function, in the OPA0 channel switching intermittent, analog comparator output will be held here to avoid
The interference caused by the unstable output during OPA0 switching. Behind this hold circuit, it is a pair of output for the comparator
With the width of the filter circuit, the user can register configuration, filter out the change after the stability of the smaller interference signal. filter
The clock signal used by the booster can be from the system clock, or 32 minutes for the internal 32MHz RC clock frequency
Frequency (1MHz). The user can select the appropriate filtering parameters according to the interference characteristics of the application environment.

Comparator output filter timing

AC0 output filter is enabled by the ACFEN bit of the OP0CRA register. The filter clock is passed through the TCKCSR register
The AFCKS bit is selected and the width of the filter can be set by the AFTCNT register. Please refer to this section for details
Definition part.

Comparator output and PWM control

LGT8FX8D series can output up to six-channel PWM signal, which by the Timer0 and Timer1 generated PWM
The signal can be used with the op amp / comparator module. The op amp / comparator output can be used to directly turn off the PWM signal,
In order to achieve a more flexible PWM protection program.

Op amp input has two channels, you can choose through the software configuration of any one channel or two channels at the same time as a control
PWM output signal source. Refer to Timer Counter 0/1 for the definition of the DSX0 / 1 register.

If the op amp function is not required in the application, the output of the comparator can also be used as a control to turn off the PWM output
Signal, so that the other channels of the comparator can be used to control the PWM output. At this point through the DSX0 / 1 need to send
The register enables the PWM output control function of the comparator op amp channel 0.

Internal reference with 8-bit digital-to-analog conversion (DAC0)

The LGT8FX8D family integrates a calibratable reference voltage source with an output voltage of 1.25V and 2.56V configurable;
This internal reference voltage provides a reference voltage source for the ADC and the analog comparator.

Inside the analog comparator, an 8-bit precision digital-to-analog converter (DAC) is integrated, and the digital-to-analog converter is
The internal reference voltage is the reference source, producing a maximum of 256 output voltages. The DAC output can be used as a negative input for analog comparators
Can also be output to the chip pin as an external reference voltage. When using the DAC's output drive other peripheral circuits
When an external voltage follower is required. The DAC output is controlled by the DACEN0 / 1 bits of the IOCR register, respectively, and the DAC outputs
The voltage is controlled by the DALR0 register. Refer to the Register Definitions section of this chapter for detailed definitions.

The DALR0 register defines the relationship with the DAC output voltage:

DALR0	DAC0 output voltage
0x00	IVREF / 256
0x01	2 * IVREF / 256
0x02	3 * IVREF / 256
...	...
...	...
0xFC	253 * IVREF / 256
0xFD	254 * IVREF / 256

0xFE	255 * IVREF / 256
0xFF	IVREF

Register definition

OPA0 control register - OP0CRA

OPA0 control register A

OP0CRA: 0x58			Default: 0x00
R / W			R / W
Initial			0x00

Bit definition

[0]	CH0EN	OPA0 Channel 0 enable control, 1: enable
[1]	CH1EN	OPA0 Channel 1 enable control, 1: enable
[2]	CH0IM	channel 0 reverse input mode enabled, 1: reverse input, 0: forward input
[3]	CH1IM	channel 1 reverse input mode enabled, 1: reverse input, 0: positive input
[4]	-	Keep it
[5]	ACFEN	Enable the analog comparator output filter function, 1: enable Refer to the Analog Comparator section for the filtering function of the analog comparator output
[6]	ACCH	Read back the currently selected OPA channel Write 1 to perform a channel switch
[7]	OPAEN	Enable OPA0 module, OPAEN set to 1, OPA0 analog front end into working condition, the user also Need to configure CH0EN / CH1EN, in order to make OPA0 correct work, the following is OPA0 working mode Definition:

CH0EN	CH1EN	OPAEN	Function Descriptions
0	0	1	Channel 0 works independently
0	1	1	Channel 1 works independently
1	0	1	Channel 0 works independently
1	1	1	Dual channel mode of operation
x	x	0	OPA0 stops working

OPA0 internal timer control register - OP0CRB

OPA0 Timer Control Register B

OP0CRB: 0x59			Default: 0x07
R / W			R / W
Initial			0x00

Bit definition

[6: 0]	TCSH	T CSH timing generation control, used to set the number of T CSH cycles, can be set to 0, prohibit the production T CHV timing, so the channel switching will be completely controlled by the timing of the T CSV Timer enable control, 1: enable timer and timer interrupt function;

[7]	OPTEN	In the OPA0 dual-channel operation mode, the timer will be automatically enabled, but will not enable the timing interrupt can

OPA0 channel switching timing control register - OP0TCNT

OPA0 channel switching timing control register

OP0TCNT / GPIOR4: 0x5a		Default: 0x00
R / W		R / W
Initial		0x00
Bit definition		
[7: 0]	OP0TCNT GPIOR4	T csv timing generation control, used to set the number of T csv cycles; when the counter count is reached After OP0TCNT, the channel is switched on and the counter is cleared again. When the OPA0 module is not enabled, this register can be used as a general purpose I / O register Use, can be used to temporarily store a byte of user data, timing fast read and write access

AC0SR - AC0 control and status registers

AC0SR - AC0 control and status registers

Address: 0x50					Default: 0x80			
Bit	7	6	5	4	3	2	1	0
Name	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
R / W	R / W	R / W	R	R / W	R / W	R / W	R / W	R / W
Initial	1	0	0	0	0	0	0	0

Bit	Name	description
7	ACD	Analog Comparator Disable bit. When the ACD bit is set to "1", the analog comparator is turned off. When the ACD bit is set to "0", the analog comparator is turned on.
6	ACBG	Analog comparator reference voltage selection control bit. When the ACBG bit is set to "1", the positive reference selects the internal reference as an input. When the ACBG bit is set to "0", the positive pin selects the external pin AIN0 as input.
5	ACO	Analog comparator output status bit. The analog comparator output is connected directly to the ACO bit after synchronization. The software can read the value of the ACO bit Gets the output value of the analog comparator.
4	ACI	Analog comparator interrupt flag bit. The ACI bit is set when the analog comparator output event triggers the interrupt mode defined by the ACIS bit. The interrupt is generated when the interrupt enable bit ACIE is "1" and the global interrupt is set. Execute the analog comparator When the service routine is serviced, the ACI will be cleared automatically, or the ACI bit will be cleared by writing "1".
3	ACIE	Analog comparator interrupt enable bit. When the ACIE bit is set to "1" and the global interrupt is set, the analog comparator interrupt is enabled. When the ACIE bit is set to "0", the analog comparator interrupt is disabled.
2	ACIC	Analog Comparator Input Capture Enable bit. When the ACIC bit is set to "1", the input capture source of the timer counter 1 is derived from the analog comparator output ACO. When the ACIC bit is set to "0", the input capture source of timer counter 1 is from external pin ICP1.
1	ACIS1	Analog Comparator Interrupt Mode Control High.

Page 195

		ACIS1 and ACIS0 together constitute ACIS [1: 0], used to control the analog comparator interrupt trigger mode.
0	ACIS0	Analog Comparator Interrupt Mode Control Low.
		ACIS0 and ACIS1 together constitute ACIS [1: 0], used to control the analog comparator interrupt trigger mode.

ACIS [1: 0]	Interrupt mode
0	The rising or falling edge of ACO triggers
1	Keep it.
2	The falling edge of ACO triggers
3	The rising edge of ACO triggers

ADCSRB - **ADC** CONTROL AND STATUS REGISTER **B**

ADCSRB - **ADC** CONTROL AND STATUS REGISTER **B**

Address: 0x7B Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	ACME01	ACME00	ACME11	ACME10		ACTS	ADTS2	ADTS1	ADTS0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	ACME01	Comparator 0 Negative input selection
6	ACME00	00: Negative Select External Input ACIN0
		01: Negative Select ADC multiplexed output
		1X: Negative Selects the output of op amp 0
5	ACME11	Comparator 1 Negative input selection
4	ACME10	00: Negative Select External Input ACIN2
		01: Negative Select ADC multiplexed output
		1X: Negative Selects the output of op amp 1
3	ACTS	AC trigger source channel selection
		0 - AC0 output as the ADC auto-conversion trigger source
		1 - AC1 output as the ADC automatic conversion trigger source
2: 0	ADTS	See ADC Register Description.

AFTCNT0 - Comparator 0 Filter Width Configuration Register

AFTCNT0 - Filter width configuration register

Address: 0x51 Default: 0xFF

Bit	AFTCNT0 [7: 0]
Name	AFTCNT0
R / W	R / W
Initial	0xFF

Bit	Name	description

Page 196

7: 0	AFCTR0	Configure the filter acquisition cycle setting, the comparator output must remain set to the length of the cycle in order to be considered

The valid comparator changes, otherwise the change will be filtered out.

The filtered clock is derived from the system clock or the internal 12KHz RC oscillator

DALR0 - DAC0 output voltage control register

DALR0 - **DAC0** output voltage control register

Address: 0x52	Default: 0xFF
Bit	AFTCNT [7: 0]
Name	AFTCNT
R / W	R / W
Initial	0xFF

Bit	Name	description
7: 0	DAL0	DAC0 output voltage control 0x00: DAO1 = IVREF / 256 0x01: DAO1 = 2 * IVREF / 256 ... 0xFE: DAO1 = 255 * IVREF / 256 0xFF: DAO1 = IVREF

DIDR1 - Digital input disable control register **1**

DIDR1 **-** Digital input disable control register **1**

Address: 0x7F					Default: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	OPAD3	OPAD2	OPAD1	OPAD0	AIND3	AIND2	AIND1	AIND0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	OPAD3	OPA3 pin digital input disable control bit
6	OPAD2	OPA2 pin digital input disable control bit
5	OPAD1	OPA1 Pin Digital Input Disable Control bit
4	OPAD0	OPA0 pin digital input disable control bit
3	AIND3	ACIN3 pin digital input disable control bit
2	AIND2	ACIN2 pin digital input disable control bit
1	AIND1	ACIN1 Pin Digital Input Disable Control bit. When the AIND1 bit is set to "1", the digital input of pin AIN1 is disabled and remains zero. when When the analog comparator is enabled, the digital input function of AIN1 is not required, so AIND1 should be set. When the AIND1 bit is set to "0", the digital input of pin AIN1 is enabled and the signal on pin Input to the internal digital logic, then set the ACD bit, that is, turn off the analog comparator.

0	AIND0	ACIN0 pin digital input disable control bit. When the AIND0 bit is set to "1", the digital input of pin AIN0 is disabled and remains zero. when When the analog comparator is enabled, the digital input function of AIN0 is not required, so AIND0 should be set. When the AIND0 bit is set to "0", the digital input of pin AIN0 is enabled and the signal on the pin can be Input to the internal digital logic, then set the ACD bit, that is, turn off the analog comparator.

Op amp / comparator *1*

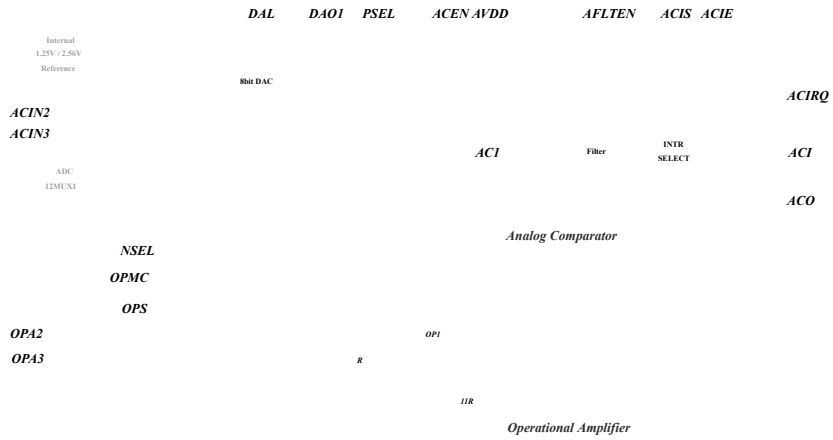
- 12mV more accurate
- Supports 2 off-chip analog inputs
- Supports multiplexing output from ADC
- Supports output from internal OPAMP op amp
- Supports internal 1.25 / 2.56V reference voltage input
- Internal integrated 8-bit DAC analog-to-digital converter
- With the pre-amplifier circuit to achieve over-current / over-voltage protection
- Programmable output filter control

Summary

The op amp / comparator module integrates an input mode configurable front stage op amp and a built-in 8-bit DAC analog ratio Compared to the front; op amp support can be configured forward / reverse input mode, fixed forward 12 times, reverse 11 times the amplification gain, Can be combined with analog comparator 8-bit DAC, to achieve a flexible signal detection; op amp 0 output can also be through the modulus Converter (ADC) for more accurate processing; analog comparator on the positive value and the value of the negative comparison, when the positive pole

Is higher than the voltage on the negative pole, the output of the analog comparator ACO is set. When the level of ACO changes,
The edge of the signal can be used to trigger an interrupt. The output signal ACO can also be used to trigger the input capture and pairing of the timer counter 1
The PWM output generated by the timer is controlled. The analog comparator integrates an 8-bit precision digital-to-analog converter (DAC)
The internal reference voltage can be decomposed into 256 different reference voltage levels.

The structure of the op amp / analog comparator 1 is shown in the following figure.



Op amp / comparator 1 module structure diagram

Input of the front op amp

The OPA1 analog front end is responsible for the multiplexed input of the analog input channels, the forward / reverse input mode switching, and the input signal
Do magnification. The OPA1 amplified signal is connected to the negative input of the analog comparator (AC).

OPA1 analog all part of the OPA1 controller under the control of the work; OPA1 analog part of the switch input or change
Input mode, the output has a stable process, OPA1 controller is responsible for generating the controller timing, to avoid not
Stable output has an effect on the post-stage circuit.

The preamplifier supports two analog input channels: OPA2 / 3. Two channels can be switched through software, can also pass
The OPA controller inside the chip automatically switches from time to time. The channel switching time has an 8-bit timer individually controlled,
Timer clock source can be selected for the system clock or internal RC32M 32-way (1MHz), to achieve more flexible
Of the channel switching cycle to meet the needs of different applications.

When the op amp function is enabled, it is necessary to set the control bits associated with the op amp pin input of the DIDR1 register to avoid I / O numbers
The functional part has an effect on the op amp's analog input channel. For details, refer to the Register Definitions section of this chapter.

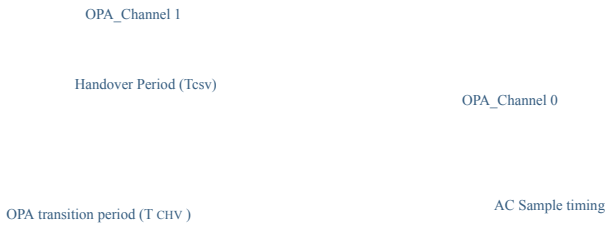
Opener channel switching mechanism

In the automatic timing channel switching mode, in order to avoid the interference generated during the channel switching process, the channel switching action and simulation
Comparator output filter linkage, to achieve a smooth channel switch. Users can also adjust the associated registers to achieve
Complete control of the switching process.

OPA1 channel switching state machine is responsible for real-time switching input channel, you can only one op amp on the basis of two
The effect of the road input. OPA1 switches the channel and switches the input mode corresponding to the channel. So we can
To achieve two positive input, two reverse input and all the way forward all the way to the three input combinations.

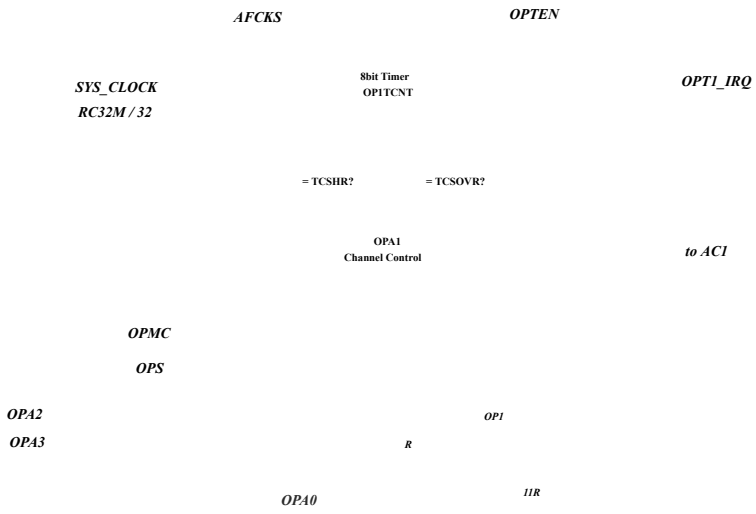
OPA1 in the process of switching the channel, the op amp output will have an unstable change, the channel switching status opportunities in the
The switching channel simultaneously generates a control signal which is used to control the next stage circuit for processing the op amp output.
Such as analog comparators, analog comparators can use this control information to its output to do the appropriate filtering process, so as to avoid
Road switching produces instability and interference.

Channel switching timing:



As shown above, T CSV is the OPA1 channel switching cycle, OPA1 start channel switching, there is an inherent build time (T CHV), during which time the output of OPA1 may be unstable or can not immediately respond to the current state of the channel, By configuring the T CHV cycle, let the next level filter out the interference introduced by the switching channel.

Page 200



OPA1 channel switching state machine contains an internal 8-bit counter, used to set the channel switching cycle (T CSV), The count clock can be the default system clock, or you can select 32 of the 32MHz RC oscillator on the chip (1 MHz). So that the user can choose the appropriate switching cycle according to the specific application requirements. For some reaction to trigger Requires fast applications, such as overcurrent protection, you can choose a faster count clock, configure a smaller counting cycle. In addition to the timer Used to generate T CSV , also used to generate T CHV cycles. The user can control the timing of channel switching through registers.

The channel switching timer, in addition to the timing used to generate channel switching, can also be used as a separate 8-bit timer Use, the timer overflow after the interrupt signal. The user can switch the input channel using the software by interrupting the service. in order to So that the software switch channel is more convenient, OPA1 controller to achieve a software switching channel dedicated control register (OP1CRA) The user only needs to write 1 to the SCSW bit in the register to implement a channel switch. When using the software timer to switch Channel, you can still configure the OP1CRB to achieve T CHV timing control.

The user can also use the channel switching timer as a timer that is completely independent of the OPA1 controller. Other systems require timing tasks. The software only needs to be set by setting the OPTEN bit of the OP1CRB register Enable timer, timer count clock selection is also a TCKCSR register AFCKS bit control, timer overflow The preset is set by the OP1TCNT register.

Analog comparator input

Both inputs of the analog comparator support a variety of optional input sources. The positive input supports the off-chip pins ACIN2 and
The 8-bit DAC, based on an internal reference, is selected by the ACBG in the ACSR of the AC control status register
Bit to control, see the register description. The negative input supports the off-chip pin ACIN3, the output of the ADC multiplexer, and from
Op amp OPA1 output. The comparator negative input channel is selected by the ADCSRB register from the ADC module
ACME10 / 11 bits to control. When ACIN1 is selected for configuration, the AIND1 bit of the DIDR1 register must be set at the same time,
Otherwise the input of ACIN1 will not be selected for the negative input of the analog comparator.

The following table shows the input control table for the analog comparator.

AC1 negative input control

ACME11	ACME10	DIDR1 [3]	MUX [2: 0]	AC Negative Input
0	0	0	xxx	ADC [xxx]
0	0	1	xxx	AIN3
0	1	1	xxx	ADC [xxx]
0	1	x	000	ADC0
0	1	x	001	ADC1
0	1	x	010	ADC2
0	1	x	011	ADC3
0	1	x	100	ADC4
0	1	x	101	ADC5
0	1	x	110	ADC6
0	1	x	111	ADC7
1	x	x	x	OPA1

Comparator output filter

Analog comparator real-time response to the input signal changes, when the input signal on the interference, the same analog comparator
The output will appear similar to the interference or instantaneous transition. This interference may result in erroneous operation. So in the simulation
Comparator output, in series with a filter circuit; filter circuit is divided into two parts, first with the op amp OPA1 with
Use the output hold circuit, in the OPA1 output analog comparator input input, if the open OPA1
The automatic channel switching function, in the OPA1 channel switching intermittent, analog comparator output will be held here to avoid
The interference caused by the unstable output during OPA1 switching. Behind this hold circuit, it is a pair of output for the comparator
With the width of the filter circuit, the user can register configuration, filter out the change after the stability of the smaller interference signal. filter
The clock signal used by the booster can be from the system clock, or 32 minutes for the internal 32MHz RC clock frequency
Frequency (1MHz). The user can select the appropriate filtering parameters according to the interference characteristics of the application environment.

Comparator output filter timing

AC1 output filter is enabled by the ACFEN bit of the OP1CRA register. The filter clock is passed through the TCKCSR register
The AFCKS bit is selected and the width of the filter can be set by the AFTCNT register. Please refer to this section for details
Definition part.

Comparator output and PWM control

LGT8FX8D series can output up to six-channel PWM signal, which by the Timer0 and Timer1 generated PWM
The signal can be used with the op amp / comparator module. The op amp / comparator output can be used to directly turn off the PWM signal,
In order to achieve a more flexible PWM protection program.

Op amp input has two channels, you can choose through the software configuration of any one channel or two channels at the same time as a control
PWM output signal source. Refer to Timer Counter 0/1 for the definition of the DSX0 / 1 register.

If the op amp function is not required in the application, the output of the comparator can also be used as a control to turn off the PWM output
Signal, so that the other channels of the comparator can be used to control the PWM output. At this point through the DSX0 / 1 need to send
The register enables the PWM output control function of the comparator op amp channel 0.

Internal reference with 8-bit digital-to-analog conversion (DAC)

The LGT8FX8D family integrates a calibratable reference voltage source with an output voltage of 1.25V and 2.56V configurable;
This internal reference voltage provides a reference voltage source for the ADC and the analog comparator.

Inside the analog comparator, an 8-bit precision digital-to-analog converter (DAC) is integrated, and the digital-to-analog converter is
The internal reference voltage is the reference source, producing a maximum of 256 output voltages. The DAC output can be used as a negative input for analog comparators
Can also be output to the chip pin as an external reference voltage. When using the DAC's output drive other peripheral circuits
When an external voltage follower is required. The DAC output is controlled by the DACEN0 / 1 bits of the IOCR register, respectively, and the DAC outputs
The voltage is controlled by the DALR1 register. Refer to the Register Definitions section of this chapter for detailed definitions.

The DALR1 register defines the relationship with the DAC output voltage:

DALR1	DAC output voltage
0x00	IVREF / 256
0x01	2 * IVREF / 256
0x02	3 * IVREF / 256
...	...
...	...
0xFC	253 * IVREF / 256
0xFD	254 * IVREF / 256
0xFE	255 * IVREF / 256
0xFF	IVREF

Register definition

OPA1 control register - OP1CRA

OPA1 control register A

OP1CRA: 0x32						Default: 0x00
R / W						R / W
Initial						0x00

Bit definition

[0]	CH0EN OPA1 channel 0 enable control, 1: enable					
[1]	CH1EN OPA1 Channel 1 enable control, 1: enable					
[2]	CH0IM channel 0 reverse input mode enabled, 1: reverse input, 0: forward input					
[3]	CH1IM channel 1 reverse input mode enabled, 1: reverse input, 0: positive input					
[4]	-	Keep it				
[5]	ACFEN	Enable the analog comparator output filter function, 1: enable Refer to the Analog Comparator section for the filtering function of the analog comparator output				
[6]	ACCH	Read back the currently selected OPA channel Write 1 to perform a channel switch				
[7]	OPAEN	Enable OPA1 module, OPAEN set to 1, OPA1 analog front end into working condition, the user also Need to configure CH0EN / CH1EN, in order to make OPA1 correct work, the following is OPA1 working mode Definition:				
		CH0EN	CH1EN	OPAEN	Function Descriptions	
		0	0	1	Channel 0 works independently	
		0	1	1	Channel 1 works independently	
		1	0	1	Channel 0 works independently	
		1	1	1	Dual channel mode of operation	
		x	x	0	OPA1 stops working	

OPA1 internal timer control register - OP1CRB

OPA1 Timer Control Register B

OP1CRB: 0x33						Default: 0x07
R / W						R / W
Initial						0x00

Bit definition

[6: 0]	TCSH	T CSH timing generation control, used to set the number of T CSH cycles, can be set to 0, prohibit the production T CHV timing, so the channel switching will be completely controlled by the timing of the T CSV				
[7]	OPTEN	Timer enable control, 1: enable timer and timer interrupt function; In the OPA1 dual-channel operation mode, the timer will automatically enable, but will not enable the timing interrupt can				

OPA1 Channel Switching Timing Control Register - OP1TCNT

OPA1 channel switching timing control register

OP1TCNT / GPIOR1: 0x34	Default: 0x00
R / W	R / W
Initial	0x00

Bit definition		
[7: 0]	OP1TCNT GPIOR1	T csv timing generation control, used to set the number of T csv cycles; when the counter count is reached OP1TCNT, start the channel switch, while the counter cleared to start again. When the OPA0 module is not enabled, this register can be used as a general purpose I / O register Use, can be used to temporarily store a byte of user data, timing fast read and write access

AC1SR - **AC1** control and status register

AC1SR - **AC1** control and status register

Address: 0x2F Default: 0x80

Bit	7	6	5	4	3	2	1	0
Name	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
R / W	R / W	R / W	R	R / W	R / W	R / W	R / W	R / W
Initial	1	0	0	0	0	0	0	0

Bit	Name	description
7	ACD	Analog Comparator Disable bit. When the ACD bit is set to "1", the analog comparator is turned off. When the ACD bit is set to "0", the analog comparator is turned on.
6	ACBG	Analog comparator reference voltage selection control bit. When the ACBG bit is set to "1", the positive reference selects the internal reference as an input. When the ACBG bit is set to "0", the positive pin selects the external pin AIN0 as input.
5	ACO	Analog comparator output status bit. The analog comparator output is connected directly to the ACO bit after synchronization. The software can read the value of the ACO bit Gets the output value of the analog comparator.
4	ACI	Analog comparator interrupt flag bit. The ACI bit is set when the analog comparator output event triggers the interrupt mode defined by the ACIS bit. The interrupt is generated when the interrupt enable bit ACIE is "1" and the global interrupt is set. Execute the analog comparator When the service routine is serviced, the ACI will be cleared automatically, or the ACI bit will be cleared by writing "1".
3	ACIE	Analog comparator interrupt enable bit. When the ACIE bit is set to "1" and the global interrupt is set, the analog comparator interrupt is enabled. When the ACIE bit is set to "0", the analog comparator interrupt is disabled.
2	ACIC	Analog Comparator Input Capture Enable bit. When the ACIC bit is set to "1", the input capture source of the timer counter 1 is derived from the analog comparator output ACO. When the ACIC bit is set to "0", the input capture source of timer counter 1 is from external pin ICP1.
1	ACIS1	Analog Comparator Interrupt Mode Control High.

0	ACIS0	ACIS1 and ACIS0 together constitute ACIS [1: 0], used to control the analog comparator interrupt trigger mode. Analog Comparator Interrupt Mode Control Low. ACIS0 and ACIS1 together constitute ACIS [1: 0], used to control the analog comparator interrupt trigger mode.

ACIS [1: 0]	Interrupt mode
0	The rising or falling edge of ACO triggers
1	Keep it.
2	The falling edge of ACO triggers
3	The rising edge of ACO triggers

ADCSRB - **ADC** CONTROL AND STATUS REGISTER **B**

ADCSRB - **ADC** CONTROL AND STATUS REGISTER **B**

Address: 0x7B Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	ACME01	ACME00	ACME11	ACME10	ACTS	ADTS2	ADTS1	ADTS0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	ACME01	Comparator 0 Negative input selection
6	ACME00	00: Negative Select External Input ACIN0
		01: Negative Select ADC multiplexed output
		1X: Negative Selects the output of op amp 0
5	ACME11	Comparator 1 Negative input selection
4	ACME10	00: Negative Select External Input ACIN2
		01: Negative Select ADC multiplexed output
		1X: Negative Selects the output of op amp 1
3	ACTS	AC trigger source channel selection
		0 - AC0 output as the ADC auto-conversion trigger source
		1 - AC1 output as the ADC automatic conversion trigger source
2: 0	ADTS	See ADC Register Description.

AFTCNT1 - Comparator 1 Filter Width Configuration Register

AFTCNT - Filter width configuration register

Address: 0x30 Default: 0xFF

Bit	AFTCNT1 [7: 0]
Name	AFTCNT1
R / W	R / W
Initial	0xFF

Bit	Name	description

7: 0	AFCTR1	Configure the filter acquisition cycle setting, the comparator output must remain set to the length of the cycle in order to be considered
		The valid comparator changes, otherwise the change will be filtered out.
		The filtered clock is derived from the system clock or the internal 12KHz RC oscillator

DALR1 - DAC1 output voltage control register

DALR **-1 DAC1** output voltage control register

Address: 0x31 Default: 0xFF

Bit	DALR1 [7: 0]
Name	DALR1
R / W	R / W
Initial	0xFF

Bit	Name	description
7: 0	DALR1	Set DAC1 voltage output control
		0x00: DAO1 = IVREF / 256
		0x01: DAO1 = 2 * IVREF / 256
		...
		0xFE: DAO1 = 255 * IVREF / 256
		0xFF: DAO1 = IVREF

DIDR1 - Digital input disable control register **1**

DIDR1 - Digital input disable control register **1**

Address: 0x7F Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	OPAD3	OPAD2	OPAD1	OPAD0	AIND3	AIND2	AIND1	AIND0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

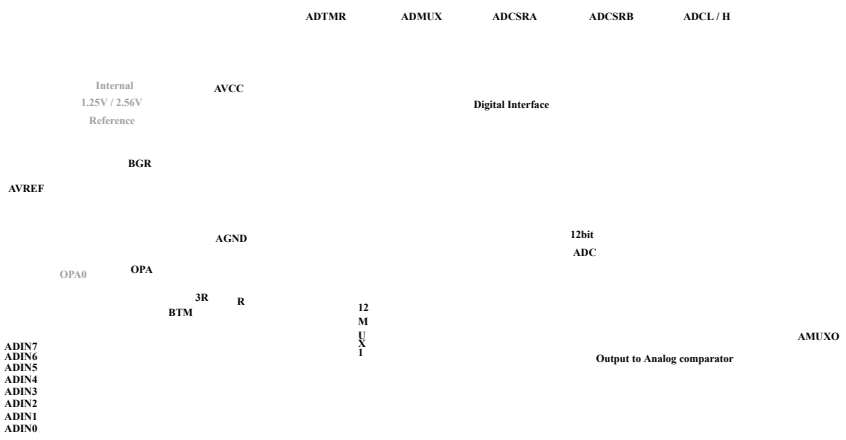
Bit	Name	description
7	OPAD3	OPA3 pin digital input disable control bit
6	OPAD2	OPA2 pin digital input disable control bit
5	OPAD1	OPA1 Pin Digital Input Disable Control bit
4	OPAD0	OPA0 pin digital input disable control bit
3	AIND3	ACIN3 pin digital input disable control bit
2	AIND2	ACIN2 pin digital input disable control bit
1	AIND1	ACIN1 Pin Digital Input Disable Control bit. When the AIND1 bit is set to "1", the digital input of pin AIN1 is disabled and remains zero. when When the analog comparator is enabled, the digital input function of AIN1 is not required, so AIND1 should be set. When the AIND1 bit is set to "0", the digital input of pin AIN1 is enabled and the signal on pin

		Input to the internal digital logic, then set the ACD bit, that is, turn off the analog comparator.
0	AIND0	ACIN0 pin digital input disable control bit. When the AIND0 bit is set to "1", the digital input of pin AIN0 is disabled and remains zero. when When the analog comparator is enabled, the digital input function of AIN0 is not required, so AIND0 should be set. When the AIND0 bit is set to "0", the digital input of pin AIN0 is enabled and the signal on the pin can be Input to the internal digital logic, then set the ACD bit, that is, turn off the analog comparator.

ADC analog - to - digital converter

- 12-bit resolution, DNL is ± 1LSB, INL is ± 1.5LSB
- Sampling rate up to 250KSPS at maximum resolution
- 8 multiplexed single-ended input channels
- The ADC input voltage range is 0-VCC
- Continuous conversion or single conversion mode
- Optional internal 1.25V / 2.56V reference voltage
- Supports AVCC and external reference voltage input
- 1 / 4VCC power supply voltage detection channel
- Input channel for internal op amp 0
- Automatic start conversion mode based on interrupt source
- The conversion result supports the optional left-justified mode
- ADC conversion end interrupt

Summary



ADC structure diagram

The analog-to-digital converter is a 12-bit successive approximation ADC. The ADC is connected to an 8-channel analog multiplexer Sampling from port A's 8 single-ended input voltages, the single-ended input voltage is based on 0V (GND).

ADC operation

The ADC converts the input analog voltage into a 10-bit digital quantity by successive approximation. The minimum value represents GND,
The maximum value represents the reference voltage minus 1LSB. The reference voltage source can be the ADC's supply voltage AVCC, the external reference voltage
AREF or internal 1.25V / 2.56V reference voltage is selected by writing the REFS bit of the ADMUX register.

The analog input channels can be selected by writing the MUX bits of the ADMUX register. Any ADC input pin, external base

Page 209

The quasi-voltage pin, and the internal reference voltage source, can be used as a single-ended input to the ADC. ADC input pin 0-5 can be used as
ADC differential input. The differential gain can be selected by writing the GAIN bit of the ADTMR register.

The ADEN bit of the ADCSRA register can be used to start the ADC. When the ADEN is cleared, the ADC does not consume power.
Turn on the ADC before entering sleep mode.

The ADC conversion result is 12 bits, which are stored in the ADC data registers ADCH and ADCL. By default, the conversion result is
Right-aligned, but can be left-aligned by setting the ADLAR bit in the ADMUX register.

If you set the conversion result to the left justify, and the maximum requires only 8 bits of conversion accuracy, then just read the ADCH is enough
The. Otherwise read the ADCL, and then read the ADCH, to ensure that the contents of the data register is the result of the same conversion.
Once ADCL is read, the data registers ADCL and ADCH are latched and the ADCH conversion result can be updated to
Data registers ADCL and ADCH.

The end of the ADC conversion can trigger an interrupt. Even if the end of the conversion occurs between reading ADCL and ADCH, the interrupt will still fire.

Start a conversion

Setting the ADSC bit to the ADC bit to write "1" can initiate a single conversion. This bit remains high during conversion
The hardware is cleared after the conversion is complete. If the channel is changed during the conversion process, the ADC will do this before changing the channel
A conversion.

ADC conversion has a different trigger source. Setting the ADCSRA register's ADC auto-enable enable bit ADATE can be enabled
Move trigger. Setting the ADCSRB register's ADC trigger selection bit ADTS can select the trigger source. When the selected trigger signal
When a rising edge is generated, the ADC prescaler is reset and begins to convert. This provides a transition at a fixed time interval
method. Even if the trigger signal persists after the conversion, a new conversion will not be initiated. If touched during the conversion process
The signal also produces a rising edge, the rising edge will also be ignored. Even if a specific interrupt is disabled or a global interrupt is made
The bit is "0" and its interrupt flag will still be set. This triggers a transition without generating an interrupt. But for
The new interrupt must be triggered when the next interrupt event occurs, and the interrupt flag must be cleared.

Using the ADC interrupt flag as the trigger source, you can start the next ADC conversion after the current conversion is complete. The
After the ADC will work in continuous conversion mode, continuous sampling and ADC data register to update. First turn
Is changed by writing "1" to the ADSCRA register ADSC bit. In this mode, subsequent ADC conversions are not followed
Depending on whether the ADC interrupt flag ADIF is set.

If automatic triggering is enabled, the ADSC setting the ADCSRA register will initiate a single conversion. The ADSC logo can also be used for inspection
Whether the conversion is in progress. Regardless of how the conversion was initiated, ADSC was always "1" during the conversion process.

Prescaler and ADC conversion timing

Under the default conditions, the successive approximation circuit requires an input clock from 300 kHz to 3 MHz for maximum accuracy. Such as
If the required conversion accuracy is less than 12 bits, then the input clock frequency can be higher than 3MHz, in order to achieve a higher sampling rate.

The ADC module includes a prescaler that can generate an acceptable ADC input clock from the system clock. Prescaler
The ADPS bits are set by the ADCSRA register. Setting ADEN of the ADCSRA register will enable ADC, pre-division

Page 210

The frequency counting starts. As long as the ADEN bit is "1", the prescaler continues counting until ADEN is cleared.

ADSCRA register ADSC is set, the single-ended conversion starts at the rising edge of the next ADC clock cycle. positive
A constant conversion requires 15 ADC clock cycles. ADC enable (ADEN of the ADCSRA register is set) requires 50
The ADC input clock cycle initializes the analog circuit before it can be effectively converted for the first time.

During ADC conversion, the sample is held at 1.5 ADC input clocks after the conversion is started, and the first ADC
The result of the conversion is output after the start of the 14.5 ADC input clock. After the conversion, the ADC result is fed
ADC data register, and the ADIF flag is set. ADSC is cleared at the same time. The software can then set ADSC again
Flag or auto trigger to start a new conversion.

Change the channel or reference source

MUX and REFS in the ADMUX register are individually buffered through temporary registers. The CPU can be used for temporary registers
Random access. Before the switch is started, the CPU can configure the channel and reference source at any time. In order to ensure that the ADC has
Adequate sampling time, once the conversion starts, do not allow the channel and the choice of the source configuration. After the conversion is complete (ADCSRA
Register ADIF is set), the channel and reference source selection will be updated. The start time of the conversion is set for ADSC
After the next ADC enters the rising edge of the clock. Therefore, it is recommended that the user enter an ADC input after ADSC
Do not operate ADMUX to select a new channel and reference source during clock cycles.

When using auto-triggering, the time at which the event occurred is uncertain. In order to control the impact of the new settings on the conversion, the update
ADMUX register should be particularly careful. If both ADATE and ADEN are set, the interrupt time can occur at any time,
Which automatically trigger, start the ADC conversion. If the contents of the ADMUX register are changed during this period, then the user is not available
The next step is based on the old configuration or the new configuration. It is recommended that the user make ADMUX at the following safety times
Update:
1) ADATE or ADEN bit is "0";
2) during the conversion process, but at least one ADC input clock cycle after the trigger event occurs;
3) After the conversion is complete, but before the interrupt flag of the trigger source is cleared.
If you update ADMUX in any of the cases mentioned above, the new configuration will take effect before the next conversion.

When selecting the ADC input channel, note that the channel is selected before the conversion is initiated and an ADC input after ADSC is set
After entering the clock cycle, you can select a new analog input channel, but the easiest way is to wait until the conversion is complete and then change
aisle.

The reference voltage source V_{ref} of the ADC reflects the conversion range of the ADC. If the single-ended channel level exceeds V_{ref}, the conversion result will be
Close to maximum 0xFFF. V_{ref} can be AVCC, the voltage of the external AREF pin, the internal 1.25V or 2.56V reference
source. Use internal reference ***(1.25V / 2.56V)*** Note:

After the chip is powered on, the internal reference is calibrated to ***1.25V by*** default . If the user uses a ***1.25V*** internal reference,
Then use, no other operation. But if you need to use the ***2.56V*** internal reference voltage, you need to update the internal benchmark
Calibration value. The calibration value of ***2.56V*** is loaded into register ***VCAL2 (0xCE)*** after power-up . When the program is initialized, the ***VCAL2***
The value is read and written to the ***VCAL (0XC8)*** register to complete the ***2.56V*** calibration.

It should be noted that when the ***VCAL*** register is updated , the calibration value of ***VCAL1*** , that is, ***1.25V*** , is updated to
VCAL2 , so if you need to re-select the ***1.25V*** benchmark during subsequent use , you need to save the value of ***VCAL1*** in advance
To the variable for later use.

Page 211

Register definition

ADC register list

register	address	Defaults	description
ADCL	0x78	0x00	ADC data low byte register
ADCH	0x79	0x00	ADC data high byte register
ADCSRA	0x7A	0x00	ADC control and status register A
ADCSRB	0x7B	0x00	ADC control and status register B
ADMUX	0x7C	0x00	ADC multiple select control register
ADTMR	0x7D	0x01	ADC mode control register
DIDR0	0x7E	0x00	Digital input disable control register 0

ADCL - ADC data low byte register

ADCL - ADC data low byte register

Address: 0x78 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name0	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Name1	ADC3	ADC2	ADC1	ADC0	-	-	-	-
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	ADC [7: 0] / ADC [3: 0]	ADC data low byte register. When the ADLAR bit is "0", the ADC output data stored in the register is placed in low order, that is, ADCL For ADC [7: 0], as shown by Name0; when the ADLAR bit is "1", the ADC output data is registered The storage in the device is aligned by the high order, ie the high 4 bits of the ADCL are ADC [3: 0], the lower 4 bits are meaningless, such as Name1.

ADCH - ADC Data High Byte Register

ADCH - ADC Data High Byte Register

Address: 0x79 Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name0	-	-	-	-	ADC11	ADC10	ADC9	ADC8
Name1	ADC11	ADC10	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	ADC [11: 8] / ADC [11: 4]	ADC data low byte register. When the ADLAR bit is "0", the ADC output data is stored in the register in low order, ie ADCH

The lower 4 bits are ADC [11: 8], the upper 4 bits are meaningless, as shown by Name0; when the ADLAR bit is "1", The ADC output data is stored in the register in high order, ie ADCH is ADC [11: 4], such as Name1.

ADCSRA - ADC CONTROL AND STATUS REGISTER A

ADCSRA - ADC CONTROL AND STATUS REGISTER A

Address: 0x7A Default: 0x02

Bit	7	6	5	4	3	2	1	0

Name	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	1	0

Bit	Name	description
7	ADEN	ADC enable control bit. When the ADEN bit is set to "1", the ADC is enabled. When the ADEN bit is set to "0", the ADC is disabled.
6	ADSC	The ADC starts to convert. In the single conversion mode, ADSC is set to initiate a conversion. In continuous conversion mode, ADSC Setting will start the first conversion.
5	ADATE	ADC auto trigger enable bit. When the ADATE bit is set to "1", the auto trigger function is enabled. The trigger signal is selected to rise Turn on once. The selection of the trigger source is controlled by the ADTS of the ADCSRB register. When the ADATE bit is set to "0", the auto trigger function is disabled.
4	ADIF	ADC interrupt flag. Set ADIF when the ADC completes a conversion and updates the data register. If the ADC interrupt is enabled ADIE is "1" and the global interrupt is set, the ADC interrupt is generated. Executing an ADC interrupt clears ADIF Bit, you can also write "1" to this bit to clear.
3	ADIE	ADC interrupt enable control bit. The ADC interrupt is enabled when the ADIE bit is set to "1" and the global interrupt is set. When the ADIE bit is set to "0", the ADC interrupt is disabled.
2: 0	ADPS [2: 0]	ADC prescaler selection control bit. ADPS Selects the system clock to generate the prescaler factor for the ADC clock.

ADPS [2: 0]	Prescaler factor
0	2
1	2
2	4
3	8
4	16
5	32
6	64
7	128

ADCSRB - ADC CONTROL AND STATUS REGISTER **B**

ADCSRB - **ADC** CONTROL AND STATUS REGISTER **B**

Address: 0x7B Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	ACME01	ACME00	ACME11	ACME10		ACTS	ADTS2	ADTS1	ADTS0
R / W	R / W	R / W	R / W	R / W	W / O	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7	ACME01	Comparator 0 Negative input selection
6	ACME00	00: Negative Select External Input ACIN0 01: Negative Select ADC multiplexed output 1X: Negative Selects the output of op amp 0
5	ACME11	Comparator 1 Negative input selection
4	ACME10	00: Negative Select External Input ACIN2 01: Negative Select ADC multiplexed output 1X: Negative Selects the output of op amp 1

3	ACTS	AC trigger source channel selection 0 - AC0 output as the ADC auto-conversion trigger source 1 - AC1 output as the ADC automatic conversion trigger source

2: 0 ADTS [2: 0] ADC auto trigger source selection control bit.

When the ADATE bit is set to "1", the auto trigger function is enabled and the selection of the trigger source is controlled by ADTS.

When setting the ADATE bit to "0", the ADTS setting is invalid. The trigger signal interrupt flag is selected to rise

Turn on once. When the trigger source is cleared from an interrupt flag to the trigger source set by the interrupt flag

Will trigger the signal to produce a rising edge, if ADEN is set at this time, ADC will open a conversion.

The auto trigger function is disabled when switching to continuous conversion mode (ADTS = 0).

ADTS [2: 0]	Trigger source
0	Continuous conversion mode
1	Comparator 0/1
2	External interrupt 0
3	Timer counter 0 compare match
4	Timer counter 0 overflows
5	Timer Counter 1 Compare Match B
6	Timer counter 1 overflows
7	Timer Counter 1 Enter the capture event

ADMUX - ADC Multiplex Select Control Register

ADMUX - **ADC** Multiplex Select Control Register

Address: 0x7C Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	REFS1	REFS0	ADLAR	CHMUX4	CHMUX3	CHMUX2	CHMUX1	CHMUX0
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 6	REFS [1: 0]	Reference voltage selection control bit. By setting the REFS control bit to select the reference voltage, if the REFS setting is changed during the conversion process, It will only work until the current conversion is complete.

REFS [1: 0]	Reference voltage selection
0	AREF
1	AVCC
2	On-chip 2.56V reference
3	On-chip 1.25V reference

5	ADLAR	Conversion result left alignment enable control bit. When the ADLAR bit is set to "1", the conversion result is left aligned in the ADC data register. When the ADLAR bit is set to "0", the conversion result is right aligned in the ADC data register.

4: 0 CHMUX [4: 0] ADC input source selection control bit.

CHMUX [4: 0]	Single-ended input source

0	PC0	External input source
1	PC1	
2	PC2	
3	PC3	
4	PC4	
5	PC5	
6	PE1	
7	PE3	
8	1 / 4VCC power supply voltage detection	Internal input source
9 ~ 12	Keep it	
13	Internal op amp 0 output	
14	Internal reference voltage source	
15	GND	

ADTMR - ADC mode control register

ADTMR - ADC mode control register

Address: 0x7D Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-			ADTM
R / W	-	-	-	-	-			R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 1	-	Keep it.
0	ADTM	Test mode, the internal reference voltage is output from the AVREF port

DIDR0 - Digital input disable control register 0

DIDR0 - Digital input disable control register **0**

Address: 0x7E Default: 0x00

Bit	7	6	5	4	3	2	1	0
Name	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W	R / W
Initial	0	0	0	0	0	0	0	0

Bit	Name	description
7: 0	ADCD [7: 0]	Digital input disable control bit. When the ADCxD bit is set to "1", the digital input of the pin ADCx is disabled and remains zero. When to make ADCx digital input function is not required when simulating the comparator, so ADCxD must be set. When the ADCxD bit is set to "0", the digital input of the pin ADCx is enabled and the signal on the pin can be lost Into the internal digital logic, then clear the ADEN bit, and turn off the analog comparator.

Electrical characteristics

Absolute working environment

Operating temperature	-40 - 85C	Important note :
storage temperature	-60 - 150C	The external working environment applied when the chip is working is larger than the listed
Pin level	0V - VCC	The maximum limit may cause the chip to be physically damaged. Chip
Maximum operating voltage	6.0V	The guaranteed normal function must ensure that the working environment is given the most
I / O maximum current	30mA	Large limit condition. When the chip is working for a long time with maximum
VCC / GND maximum current	200mA	Parameters, will affect the chip life and device stability
ESD characteristics	≥ ± 4KV	Sex.

DC characteristics

Typical DC characteristics TA = -40 - + 85C, VCC = 1.8V - 5.5V

Symbol	Parameter	Condition	Min.	Typ	Max.	Unit
V_{IL}	Enter the low level threshold			VCC / 3		V
V_{IH}	Enter the high level threshold			VCC / 2		V
V_{OL}	Output low level threshold	I_{OL} = 40mA, VCC = 5V			0.8	V
		I_{OL} = 25 mA, VCC = 3.3 V			0.7	
V_{OH}	Output high level threshold	I_{OH} = 20mA, VCC = 5V	4.4V			V
		I_{OH} = 12 mA, VCC = 3.3 V	2.6V			
I_{IL}	I / O input low leakage				1	uA
I_{IH}	I / O input high leakage				1	uA
R_{WPU}	I / O weak pull-up value			80K		Ω
R_{PU}	I / O strong pull-up resistance			15K		Ω
I_{CC}	Active	1MHz@3.3V		0.56		mA
		4MHz@3.3V		1.25		
	IDLE	4MHz@3.3V		0.30		mA
	Power / Off S0	w / o WDT@3.3V		12.0		uA
	Power / Off S1	VCC = 3.3V		7.4		uA

Description:

1. The above power consumption test will be configured to drive low-drive mode, detailed configuration, please refer to the power control related documents;
2. Test code drives an I / O output with a fixed frequency square wave and cyclically reads the status of a port;

Internal clock characteristics

	Frequency	VCC	Temperature	Accuracy
Factory	32MHz	2.5V - 5V	25C	± 2%
Calibration	32KHz	2.5V - 5V	25C	± 2%

Internal reference voltage characteristics

	Voltage	VCC	Temperature	Accuracy
Factory	1.25V	2.0V - 5V	25C	± 1%
Calibration	2.56V	3.3V - 5V	25C	± 1%

Low voltage detection circuit characteristics

VDTS	Min.VBOT	Typ. VBOT	Max. VBOT	Uint
111		VDT Disabled		
110	1.6	1.7	2.0	
101	2.4	2.5	2.8	V
100	3.8	4.0	4.5	
011				
...		Reserved		
000				

ADC circuit characteristics

Symbol	Parameter	Condition	Min.	Typ	Max.	Unit
	Resoultion			12		Bits
V_{IN}	Input Voltage		GND		V_{REF}	V
AV_{CC}	Power Supply		2.0		VCC	V
V_{REF}	Reference Voltage		1.0		AV_{CC}	V
CLK	Clock Frequency		50K		10M	Hz
	Conversion Time	Free running		19		CLK
INL	Integral Non-Linearity			3		LSB
DNL	Differential Non-Linearity			3		LSB
R_{AIN}	Input Resistance			100		MΩ

Comparator circuit characteristics

Symbol	Parameter	Condition	Min.	Typ	Max.	Unit
	Resoultion			5		mV
V IN	Input Voltage		GND		VCC	V
AV CC	Power Supply		2.0		VCC	V
R AIN	Input Resistance			100		MΩ

Operational amplifier circuit characteristics

Symbol	Parameter	Condition	Min.	Typ	Max.	Unit
Gain	Amplify Gain	V IN = 100mV ~ 220mV		12		
V IN	Input Voltage				(VCC-0.9) / Gain V	
AV CC	Power Supply		2.0		VCC	V
R AIN	Input Resistance			100		MΩ

Op amp input and magnification Response characteristics:

1	2	3	4	5	6	7	8	9	10	Unit
20	40	60	80	90	100	110	120	130	140	mV

11	12	13	14	15	16	17	18	19	20	Uint
150	160	170	180	190	200	210	220	230	240	mV

Register lookup table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
				Extended IO Register					
$ F6	GUID3				GUID Byte 3				

$ F5	GUID2	GUID Byte 2							
$ F4	GUID1	GUID Byte 1							
$ F3	GUID0	GUID Byte 0							
$ F2	PMCR	PMCE	CLKFS	CLKSS	WCLKS	OSCKEN	OSCMEN	RCKEN	RCMEN
$ F1									
$ F0	IOCR	IOCE	STOSC1	STOSC0	DACEN1	DACEN0	XIEN	REFIOEN	RSTIOEN
$ EE	PMXCR					OC0C0	SSB1	TDD6	RDD5
$ EC	TKCSR	-	F2XEN	TC2XF1	TC2XF0	-	AFCKS	TC2XS1	TC2XS0
$ E2	PSSR	PSS1	-	-	-	-	-	-	PSR1
$ CF	LDOCR	WEN				PDEN	VSEL2	VSEL1	VSEL0
$ CE	VCAL2	Calibration value for 2.56V internal reference							
$ CD	VCAL1	Calibration value for 1.2V internal reference							
$ C8	VCAL	Internal Voltage Reference calibration register							
$ C6	UDR0	USART Data							
$ C5	UBRR0H	-	-	-	-	USART Baud Rate Register High			
$ C4	UBRR0L	USART Baud Rate Register Low							
$ C2	UCSR0C	UMSEL0		UPM0		USBS0	UCSZ01 / UDORD0	UCSZ00 / UCPHA0	UCPOL0
$ C1	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80
$ C0	UCSR0A	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0
$ BD	TWAMR	TWI Address Mask							-
$ BC	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE
$ BB	TWDR	TWI Data							
$ BA	TWAR	TWI Address							TWGCE
$ B9	TWSR	TWI Status					-	TWPS	
$ B8	TWBR	TWI Bit Rate							
$ B6	ASSR	-	EXCLK	AS2	TCN2UB	OCR2AUB	OCR2BUB	TCR2AUB	TCR2BUB

Page 220

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$ B4	OCR2B	Timer / Counter 2 Output Compare Register B							
$ B3	OCR2A	Timer / Counter 2 Output Compare Register A							
$ B2	TCNT2	Timer / Counter 2 Counter Register							
$ B1	TCCR2B	FOC2A	FOC2B	-	-	WGM22		CS2	
$ B0	TCCR2A	COM2A		COM2B		-	-	WGM21	WGM20
$ A9	PORTE	Port Output E							
$ A8	DDRE	Data Direction E							
$ A7	PINE	Port Input E							

$ 8B	OCR1BH	Timer / Counter 1 Output Compare B High							
$ 8A	OCR1BL	Timer / Counter 1 Output Compare B Low							
$ 89	OCR1AH	Timer / Counter 1 Output Compare A High							
$ 88	OCR1AL	Timer / Counter 1 Output Compare A Low							
$ 87	ICR1H	Timer / Counter 1 Input Capture High							
$ 86	ICR1L	Timer / Counter 1 Input Capture Low							
$ 85	TCNT1H	Timer / Counter 1 Counter High							
$ 84	TCNT1L	Timer / Counter 1 Counter Low							
$ 82	TCCR1C	FOC1A	FOC1B	-	-	-	-	-	-
$ 81	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12		CS1	
$ 80	TCCR1A	COM1A		COM1B		-	-	WGM11	WGM10
$ 7F	DIDR1	OPA3D	OPA2D	OPA1D	OPA0D	AIN3D	AIN2D	AIN1D	AIN0D
$ 7E	DIDR0	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D
$ 7D	ADTMR	-	-	-	-	-	-	-	ADTM
$ 7C	ADMUX	REFS		ADLAR	-		MUX		
$ 7B	ADCSRB	ACME01	ACME00	ACME11	ACME10		-	ADTS	
$ 7A	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE		ADPS	
$ 79	ADCH	ADC Data High							
$ 78	ADCL	ADC Data Low							
$ 75	IVBASE	Interrupt Vector Base Address							
$ 73	PCMSK3	PCINT30	PCINT29	PCINT28	PCINT27		PCINT26	PCINT25	PCINT24
$ 70	TIMSK2	-	-	-	-	-	OCIE2B	OCIE2A	TOIE2
$ 6F	TIMSK1	-	-	ICIE1	-	-	OCIE1B	OCIE1A	TOIE1
$ 6E	TIMSK0	-	-	-	-	-	OCIE0B	OCIE0A	TOIE0
$ 6D	PCMSK2	PCINT [23:16]							

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$ 6C	PCMSK1	PCINT [15: 8]							
$ 6B	PCMSK0	PCINT [7: 0]							
$ 69	EICRA	-	-	-	-	ISC1		ISC0	
$ 68	PCICR	-	-	-	-	PCIE3	PCIE2	PCIE1	PCIE0
$ 66	OSCCAL	-	-	OSC Calibration					
$ 65	PRR1	-	-	PRWDT	-	-	PREFL	PRPCI	-
$ 64	PRR	PRTWI	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	PRUSART0	PRADC
$ 62	VDTCR	VDTCE	SWRSTN	-	-	VDTSEL		LVREN	VDTEN
$ 61	CLKPR	CLKPCE	CLKOEN1	CLKOEN0	-		CLKPS		
$ 60	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0
		IO Register							
$ 5F ($ 3F)	SREG	I	T	H	S	V	N.	Z	C
$ 5E ($ 3E)	SPH	Stack Point High							
$ 5D ($ 3D)	SPL	Stack Point Low							
$ 5A ($ 3A)	OP0TCNT	OPA0 Channel Switch Periold Regiser							
$ 59 ($ 39)	OP0CRB	OPTEN	OPA1 Channel Switch Hold Timing Register						
$ 58 ($ 38)	OP0CRA	OP1EN	ACCH	ACFEN	-	CH1IM	CH0IM	CH1EN	CH0EN

$ 56 ($ 36)	ECCR	WEN	EEN	-	-	-	-	EC1	EC0
$ 55 ($ 35)	MCUCR	FWKEN	FPDEN	-	PUD	-	-	IVSEL	IVCE
$ 54 ($ 34)	MCUSR	SWDD	-	-	OCDRF	WDRF	BORF	EXTRF	PORF
$ 53 ($ 33)	SMCR	-	-	-	-		SM		SE
$ 52 ($ 32)	DAL0	DAC0 Output data Register							
$ 51 ($ 31)	AFTCNT0	AC0 Filter Timing Register							
$ 50 ($ 30)	AC0SR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS	
$ 4E ($ 2E)	SPDR	SPI Data							
$ 4D ($ 2D)	SPSR	SPIF	WCOL	-	-	-	DUAL	-	SPI2X
$ 4C ($ 2C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR	
$ 4B ($ 2B)	GPIOR2	General Purpose IO Register 2							
$ 4A ($ 2A)	GPIOR1	General Purpose IO Register 1							
$ 48 ($ 28)	OCR0B	Timer / Counter 0 Output Compare Register B							
$ 47 ($ 27)	OCR0A	Timer / Counter 0 Output Compare Register A							
$ 46 ($ 26)	TCNT0	Timer / Counter 0 Counter							
$ 45 ($ 25)	TCCR0B	FOC0A	FOC0B	OC0AS	-	WGM02		CS0	
$ 44 ($ 24)	TCCR0A	COM0A		COM0B		-	-	WGM01	WGM00

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
$ 43 ($ 23)	GTCCR	TSM	-	-	-	-	-	PSRASY	PSRSYNC
$ 42 ($ 22)	EEARH	EEPROM Address High							
$ 41 ($ 21)	EEARL	EEPROM Address Low							
$ 40 ($ 20)	EEDR	EEPROM Data Low							
$ 3F ($ 1F)	EECR	EEPM2	-	EEPM1	EEPM0	EERIE	EEMWE	EEWE	EERE
$ 3E ($ 1E)	GPIOR0	General Purpose IO Register 0							
$ 3D ($ 1D)	EIMSK	-	-	-	-	-	-	INT1	INT0
$ 3C ($ 1C)	EIFR	-	-	-	-	-	-	INTF1	INTF0
$ 3B ($ 1B)	PCIFR	-	-	-	-	PCIF3	PCIF2	PCIF1	PCIF0
$ 37 ($ 17)	TIFR2	-	-	-	-	-	OCF2B	OCF2A	TOV2
$ 36 ($ 16)	TIFR1	-	-	ICF1	-	-	OCF1B	OCF1A	TOV1
$ 35 ($ 15)	TIFR0	-	-	-	-	-	OCF0B	OCF0A	TOV0
$ 34 ($ 14)	OP1TCNT	OP1 Channel Switch Period Tming Register							
$ 33 ($ 13)	OP1CRB	OPTEN			OP1 Channel Switch Hold Timing Register				
$ 32 ($ 12)	OP1CRA	OP1EN		ACCH	ACFEN	-	CH1IM	CH0IM	CH1EN	CH0EN
$ 31 ($ 11)	DAL1	DAC1 Output level select							
$ 30 ($ 10)	AFTCNT1	AC1 Filter Timing Register							
$ 2F ($ 0F)	AC1CSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS	
$ 2B ($ 0B)	PORTD	Port Output D							
$ 2A ($ 0A)	DDRD	Data Direction D							
$ 29 ($ 09)	PIND	Port Input D							
$ 28 ($ 08)	PORTC	Port Output C							
$ 27 ($ 07)	DDRC	Data Direction C							
$ 26 ($ 06)	PINC	Port Input C							
$ 25 ($ 05)	PORTB	Port Output B							
$ 24 ($ 04)	DDRB	Data Direction B							
$ 23 ($ 03)	PINB	Port Input B							

Instruction set

instruction	Operands	description	operating	Mark bit	cycle
Arithmetic logic instructions					
ADD	R_d , R_r	Register is added	$R_d \leftarrow R_d + R_r$	Z, C, N, V, H	1
ADC	R_d , R_r	Supports with carry registers	$R_d \leftarrow R_d + R_r + C$	Z, C, N, V, H	1
ADIW	R_{dl} , K	Immediate additions to words	$R_{dh} : R_{dl} \leftarrow R_{dh} : R_{dl} + K$	Z, C, N, V, S	1
SUB	R_d , R_r	Register addition and subtraction	$R_d \leftarrow R_d - R_r$	Z, C, N, V, H	1
SUBI	R_d , K	Register decrements	$R_d \leftarrow R_d - K$	Z, C, N, V, H	1
SBC	R_d , R_r	The register with the borrow is added and subtracted	$R_d \leftarrow R_d - R_r - C$	Z, C, N, V, H	1
SBCI	R_d , K	The register with borrow is decremented	$R_d \leftarrow R_d - K - C$	Z, C, N, V, H	1
SBIW	R_{dl} , K	Immediate subtraction with words	$R_{dh} : R_{dl} \leftarrow R_{dh} : R_{dl} - K$	Z, C, N, V, S	1
AND	R_d , R_r	Logical and	$R_d \leftarrow R_d \& R_r$	Z, N, V	1
ANDI	R_d , K	Register logic and constants	$R_d \leftarrow R_d \& K$	Z, N, V	1
OR	R_d , R_r	Logical or	$R_d \leftarrow R_d \mid R_r$	Z, N, V	1
ORI	R_d , K	Register logic or constant	$R_d \leftarrow R_d \mid K$	Z, N, V	1
EOR	R_d , R_r	Register exclusive OR	$R_d \leftarrow R_d \oplus R_r$	Z, N, V	1
Com	R_d	Reverse code	$R_d \leftarrow \$ FF - R_d$	Z, C, N, V	1
NEG	R_d	2 forcibly	$R_d \leftarrow \$ 00 - R_d$	Z, C, N, V, H	1
SBR	R_d , K	Set the bits in the register	$R_d \leftarrow R_d \vee K$	Z, N, V	1
CBR	R_d , K	The bits in the clear register	$R_d \leftarrow R_d \vee (\$ FF - K)$	Z, N, V	1
INC	R_d	Increment	$R_d \leftarrow R_d + 1$	Z, N, V	1
DEC	R_d	Diminished	$R_d \leftarrow R_d - 1$	Z, N, V	1
TST	R_d	The test is 0 or negative	$R_d \leftarrow R_d \& R_d$	Z, N, V	1
CLR	R_d	Clear register	$R_d \leftarrow R_d \oplus R_d$	Z, N, V	1
SER	R_d	The register is all set to 1	$R_d \leftarrow \$ FF$	None	1
MUL	R_d , R_r	Unsigned multiplication	$R_1 : R_0 \leftarrow R_d \times R_r$	Z, C	1
MULS	R_d , R_r	Symbolic multiplication	$R_1 : R_0 \leftarrow R_d \times R_r$	Z, C	1
MULSU	R_d , R_r	Signed number of unsigned numbers	$R_1 : R_0 \leftarrow R_d \times R_r$	Z, C	1
FMUL	R_d , R_r	Unsigned multiplication	$R_1 : R_0 \leftarrow (R_d \times R_r) << 1$	Z, C	1
FMULS	R_d , R_r	Signed multiplication, shift	$R_1 : R_0 \leftarrow (R_d \times R_r) << 1$	Z, C	1
FMULSU	R_d , R_r	Signed number of unsigned numbers, shifted	$R_1 : R_0 \leftarrow (R_d \times R_r) << 1$	Z, C	1
Jump instruction					
RJMP	K	Relative jump	PC ← PC + K + 1	None	1
IJMP		Indirect jump (to Z point to address)	PC ← Z	None	2
JMP	K	Direct jump	PC ← K	None	2
RCALL	K	Relative address subroutine call	PC ← PC + K + 1	None	1
ICALL		Indirect subroutine call (Z point to address)	PC ← Z	None	2

CALL	K	Direct subroutine call	PC ← K	None	2
RET		Subroutine returns	PC ← Stack	None	2
RETI		Interrupt the return	PC ← Stack	I	2

instruction	Operands	description	operating	Mark bit	cycle
Jump instruction (continued)					
CPSE	R_d , R_r	Equal jump	If (R_d = R_r) PC ← PC + 2 or 3	None	1/2
CP	R_d , R_r	Comparison	R_d - R_r	Z, N, V, C, H	1
The	R_d , R_r	With carry comparison	R_d - R_r - C	Z, N, V, C, H	1
CPI	R_d , K	Compare with immediate data	R_d - K	Z, N, V, C, H	1
SBRC	R_r , b	The bit is 0 to skip the next instruction	If (R_r (b) = 0) PC ← PC + 2 or 3	None	1/2
SBRS	R_r , b	Bit 1 skip the next instruction	If (R_r (b) = 1) PC ← PC + 2 or 3	None	1/2
SBIC	P, b	The I / O bit is 0 to skip the next instruction	If (P (b) = 0) PC ← PC + 2 or 3	None	1/2
SBIS	P, b	The I / O bit is 1 to skip the next instruction	If (P (b) = 1) PC ← PC + 2 or 3	None	1/2
BRBS	s, k	The status flag is 1 to jump	If (SREG (S) = 1) PC ← PC + K + 1	None	1/2
BRBC	s, k	The status flag is 0 to jump	If (SREG (S) = 0) PC ← PC + K + 1	None	1/2
BREQ	k	Equal jump	if (Z = 1) then PC ← PC + k + 1	None	1/2
BRNE	k	Ranging from jumping	if (Z = 0) then PC ← PC + k + 1	None	1/2
BRCS	k	Carry is jump	if (C = 1) then PC ← PC + k + 1	None	1/2
BRCC	k	No carry jump	if (C = 0) then PC ← PC + k + 1	None	1/2
BRSH	k	Not less than the jump	if (C = 0) then PC ← PC + k + 1	None	1/2
BRLO	k	Less than jump	if (C = 1) then PC ← PC + k + 1	None	1/2
BRMI	k	Jump for negative	if (N = 1) then PC ← PC + k + 1	None	1/2
BRPL	k	For the regular jump	if (N = 0) then PC ← PC + k + 1	None	1/2
BRGE	k	Signed not to jump	if (N ⊕ V = 0) then PC ← PC + k + 1	None	1/2
BRLT	k	Signed less than 0 jumps	if (N ⊕ V = 1) then PC ← PC + k + 1	None	1/2
BRHS	k	Half carry for 1 jump	if (H = 1) then PC ← PC + k + 1	None	1/2
BRHC	k	Half carry is 0 to jump	if (H = 0) then PC ← PC + k + 1	None	1/2
BRTS	k	T is set to jump	if (T = 1) then PC ← PC + k + 1	None	1/2
BRTC	k	T is cleared	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	Overflow jumps	f (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	Do not overflow is jump	f (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	The global interrupt is enabled	f (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	The global interrupt is disabled	f (I = 0) then PC ← PC + k + 1	None	1/2
Data transfer instructions					
MOV	Rd, Rr	Move data between registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Move a word of data	Rd + 1: Rd ← Rr + 1: Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Indirect loading	Rd ← (X)	None	1/2
LD	Rd, X +	Indirect load, address increment	Rd ← (X), X ← X + 1	None	1/2
LD	Rd, -X	The address is decremented and loaded indirectly	X ← X - 1, Rd ← (X)	None	1/2
LD	Rd, Y	Indirect loading	Rd ← (Y)	None	1/2
LD	Rd, Y +	Indirect load, address increment	Rd ← (Y), Y ← Y + 1	None	1/2
LD	Rd, -Y	The address is decremented and loaded indirectly	Y ← Y - 1, Rd ← (Y)	None	1/2
LDD	Rd, Y + q	Indirect loading with offset	Rd ← (Y + q)	None	1/2

Page 225

LD	Rd, Z	Indirect loading	Rd ← (Z)	None	1/2
LD	Rd, Z +	Indirect load, address increment	Rd ← (Z), Z ← Z + 1	None	1/2
LD	Rd, -Z	The address is decremented and loaded indirectly	Z ← Z - 1, Rd ← (Z)	None	1/2
LDD	Rd, Z + q	Indirect loading with offset	Rd ← (Z + q)	None	1/2
LDS	Rd, k	Load directly from SRAM	Rd ← (k)	None	2
ST	X, Rr	Indirect storage	(X) ← Rr	None	1
ST	X +, Rr	Indirect storage, address increment	(X) ← Rr, X ← X + 1	None	1
ST	-X, Rr	The address is decremented and stored indirectly	X ← X - 1, (X) ← Rr	None	1
ST	Y, Rr	Indirect storage	(Y) ← Rr	None	1
ST	Y +, Rr	Indirect storage, address increment	(Y) ← Rr, Y ← Y + 1	None	1
ST	-Y, Rr	The address is decremented and stored indirectly	Y ← Y - 1, (Y) ← Rr	None	1
STD	Y + q, Rr	Indirect storage with offset	(Y + q) ← Rr	None	1
ST	Z, Rr	Indirect storage	(Z) ← Rr	None	1
ST	Z +, Rr	Indirect storage, address increment	(Z) ← Rr, Z ← Z + 1	None	1
ST	-Z, Rr	The address is decremented and stored indirectly	Z ← Z - 1, (Z) ← Rr	None	1
STD	Z + q, Rr	Indirect storage with offset	(Z + q) ← Rr	None	1
STS	k, Rr	Directly stored in the SRAM	(k) ← Rr	None	2
LPM		Load the program space data	R0 ← (Z)	None	2
LPM	Rd, Z	Load the program space data	Rd ← (Z)	None	2
LPM	Rd, Z +	Load the program data, the address is incremented	Rd ← (Z), Z ← Z + 1	None	2
LD	Rd, Z +	Indirect load, address increment	Rd ← (Z), Z ← Z + 1	None	1
LD	Rd, -Z	The address is decremented and loaded indirectly	Z ← Z - 1, Rd ← (Z)	None	1
LDD	Rd, Z + q	Indirect loading with offset	Rd ← (Z + q)	None	1
LDS	Rd, k	Load directly from SRAM	Rd ← (k)	None	2
IN	Rd, P	Read port	Rd ← P	None	1
OUT	P, Rr	Write port	P ← Rr	None	1
PUSH	Rr	Push the stack	STACK ← Rr	None	1
POP	Rd	Out of stack	Rd ← STACK	None	1/2
SBI	P, b	Set the IO register	I / O (P, b) ← 1	None	1
CBI	P, b	Clear the IO register	I / O (P, b) ← 0	None	1
LSL	Rd	Logic left shift	Rd (n + 1) ← Rd (n), Rd (0) ← 0	Z, C, N, V	1
LSR	Rd	Logically shifted right	Rd (n) ← Rd (n + 1), Rd (7) ← 0	Z	1
ROL	Rd	The loop containing the carry is shifted left	(0) ← C, Rd (n + 1) ← Rd (n), C ← Rd (7)	Z	1
ROR	Rd	The loop containing the carry is shifted right	(7) ← C, Rd (n) ← Rd (n + 1), C ← Rd (0)	Z	1
ASR	Rd	Arithmetic right shift	Rd (n) ← Rd (n + 1), n = 0: 6	Z	1
SWAP	Rd	Bit exchange	(3: 0) ← Rd (7: 4), Rd (7: 4) ← Rd (3: 0)	None	1
BSET	s	Set the status bit	SREG (s) ← 1	SREG (s)	1
BCLR	s	Clear status bit	SREG (s) ← 0	SREG (s)	1
BST	Rr, b	Stored in the T bit	T ← Rr (b)	T	1
BLD	Rd, b	Read the T bit into the register	Rd (b) ← T	None	1
SEC		Set the carry flag	C ← 1	C	1

Page 226

CLC	Clear carry mark	C ← 0	C	1
SEN	Set the negative flag	N ← 1	N.	1
CLN	Clear negative flag	N ← 0	N.	1

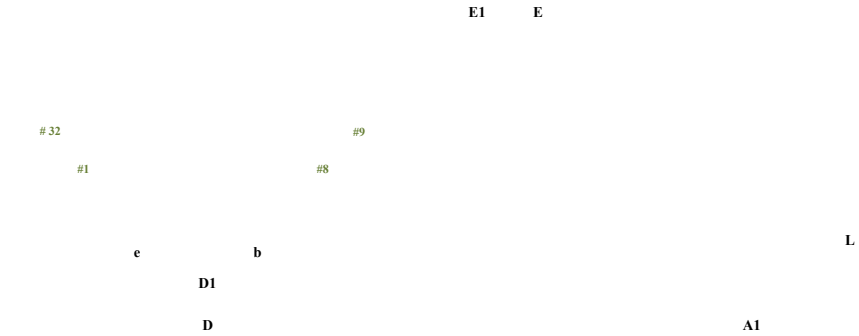
SEZ	Set the zero flag	Z ← 1	Z	1
CLZ	Clear the zero mark	Z ← 0	Z	1
SEI	Enable global interrupts	I ← 1	I	1
CLI	Disable global interrupts	I ← 0	I	1
SES	Set the symbol test flag	S ← 1	S	1
CLS	Clear the symbol test flag	S ← 0	S	1
SEV	Set the twos complement overflow flag	V ← 1	V	1
CLV	Clear the two's complement overflow flag	V ← 0	V	1
SET	Set the T bit (SREG)	T ← 1	T	1
CLT	Clear T bit (SREG)	T ← 0	T	1
MCU control instructions				
NOP	Empty command		None	1
SLEEP	Go to sleep mode		None	1
WDR	Watchdog reset		None	1
BREAK	Soft breakpoint	Only for debugging purposes	None	N / A
NOP	Empty command		None	1
SLEEP	Go to sleep mode		None	1

Encapsulation parameters

C

#twenty four **# 17**

 # 25 **# 16**



LQFP32L Universal Dimension Definitions

Character code	Minimum value	Typical value	The maximum value	unit
D	8.90	9.00	9.10	mm
D1	6.90	7.00	7.10	mm
b	0.15	0.20	0.25	mm
e	0.75	0.80	0.85	mm
E	8.90	9.00	9.10	mm
E1	6.90	7.00	7.10	mm
C	-	0.10	-	mm
L	0.55	0.60	0.65	mm
A1	-	1.40	-	mm

SSOP20L Universal Dimension Definitions

Character code	Minimum value	Typical value	The maximum value	unit
D	6.90	7.20	7.50	mm

A2	0.03	0.05	0.07	mm
b	0.22	0.30	0.38	mm
e	-	0.65	-	mm
E	7.40	7.80	8.20	mm
E1	5.00	5.30	5.60	mm
L1	0.55	-	0.95	mm
L	-	-	-	mm
A1	-	2.0	-	mm

Version history

V1.0.5	Added a description of the package pin multiplexing relationship
V1.0.4	Updated SSOP24 / 20L package pin function definition, updated part of the electrical parameters
V1.0.3	Added SSOP20L package size definition
V1.0.2	Added I / O default pull-up resistor description (input / output subsystem section p47)
	Increased use of the internal reference voltage (ADC section p204)
V1.0.1	Fixed some sections describing errors
V1.0.0	initial version
2015-02-03	