# JZ8P1510

## 8-bit OTP Microcontroller

## User Data Manual

Version V1.3

## Instructions for Modifying Records

| Version number | Modified instructions | Notes |
|---|---|---|
| V0.0 | First draft version | |
| V1.0 | Release). | |
| V1.1 | Increasing attention, low-speed oscillator error range | |
| V1.2 | Increase interrupts enabled and matters needing attention | |
| V1.3 | Add interrupt flag considerations | |
| | | |

## What to watch for:

1) Not recommended for applications with high EFT requirements or AC resistive step-down supplies.

2) The result of ISR register is no longer the value after the sum of ISR and IMR. Disabling interrupt enable does not clear the interrupt flag, the user needs to clear the flag in time when disabling interrupt enable, otherwise it will go into the interrupt all the time (the chip doesn't support to realize the operation of [Clear Interrupt Flag] through [Disable Interrupt Enable], the user needs to perform the operation of [Clear Interrupt Flag] after [Disable Interrupt Enable]).

4) Port state change wakeup is no longer required to set ICIE=1.

5) enter the interrupt, used to clear the interrupt flags and useless in clear at the time of the interruption to the logo.

# CONTENT

**1. General description ..... 5**
1.1 Functional features ..... 5
1.2 Pin assignment ..... 6
1.3 Pin description ..... 7
1.4 System block diagram ..... 8
**2. Memory structure ..... 9**
2.1 Program storage area ..... 9
2.2 The data store ..... 10
**3 Functional description ..... 11**
3.1 Operate registers ..... 11
3.1.1 RPAGE ~ R0/IAR (Indirect addressing register) ..... 11
3.1.2 RPAGE ~ R1 / TCC (Timing counter) ..... 11
3.1.3 RPAGE ~ R2/PC(Program counter) ..... 11
3.1.4 RPAGE ~ R3/STATUS(status register) ..... 12
3.1.5 RPAGE ~ R4/RSR(RAM select register) ..... 13
3.1.6RPAGE ~ R6/P0RT6(P6 data register) ..... 14
3.1.7 RPAGE ~ R8/PWMCON(PWM control register) ..... 14
3.1.8 RPAGE ~ R9/PRDL(PWM periodic low bit register) ..... 15
3.1.9 RPAGE ~ RA/PDCL(PWM duty cycle low bit register) ..... 15
3.1.10 RPAGE ~ RB/PRDCH(PWM Cycle Duty Cycle High bit register) ..... 15
3.1.12 RPAGE ~ RD/ICIECR(Input state change interrupt enabled) ..... 15
3.1.13 RPAGE ~ RE/CPUCON(CPU mode control register) ..... 16
3.1.14 RPAGE ~ RF/ISR(Interrupt flag register) ..... 17
3.2 Control registers ..... 18
3.2.1 CONT(Control register) ..... 18
3.2.2 IOPAGE ~ IOC6/P6CR(P6 direction control register) ..... 19
3.2.2 IOPAGE ~ IOC7/T1L(T1/PWM counter low bit) ..... 19
3.2.2 IOPAGE ~ IOC8/T1H(T1/PWM counter high bit) ..... 19
3.2.3 IOPAGE ~ IOC9/PDCR0(port pull-down control register 0) ..... 19
3.2.4 IOPAGE ~ IOCB/PDCR1(Port pull-down control register 1) ..... 20
3.2.5 IOPAGE ~ IOCD/PHCR(Port pull-up control register) ..... 20
3.2.6 IOPAGE ~ IOCE/EISCR(EIS Control register) ..... 20
3.2.7 IOPAGF ~ IOCF/IMR(Interrupt enable control register) ..... 20
3.3 Interrupt ..... 22
3.3.1 Interrupt site protection ..... 23
3.4 Reset ..... 24
3.4.1 Overview of reset function ..... 24
3.4.2 POR electric reset ..... 24
3.4.3 LVR low voltage reset ..... 25
3.4.4 Working frequency relationship with as low pressure detection ..... 26
3.5.1 High-speed mode ..... 28
3.5.2 Low-speed mode ..... 29

3.5.3 Idle mode ..... 29
3.5.4 Sleep patterns ..... 30
3.6 System clock ..... 31
3.6.1 Internal RC oscillator ..... 31
3.6.3 Clock module application instructions ..... 31
3.7 I/O port ..... 32
3.7.1 GPIO internal structure ..... 32
3.7.2 Port state changes ..... 32
3.7.3 Port parameters ..... 33
3.8 TCC timer counter ..... 34
3.8.1 TCC timing settings ..... 34
3.8.2 TCC showed timing calculation ..... 35
3.9 PWM pulse width modulation ..... 36
3.9.1 PWM internal structure and sequence ..... 36
3.9.2 PWM cycle and duty cycle ..... 36
3.9.3 PWM pulse width modulation setting instructions ..... 37
3.10 Wake up instructions ..... 38
**4. OPTION configuration ..... 39**
**5. Instruction set ..... 41**
**6. Electrical characteristics ..... 43**
6.1 Limit parameters ..... 43
6.2 DC electrical characteristics ..... 43
6.3 Characteristic curve diagram ..... 44
6.3.1 Internal low-speed RC Oscillator - voltage-frequency characteristic curve ..... 44
6.3.2 Internal low-speed RC oscillator - Temperature-frequency characteristic curve ..... 44
6.3.3 Internal 1MHz RC oscillator - voltage-frequency characteristic curve ..... 45
6.3.4 Internal 1MHz RC oscillator - Temperature-frequency characteristic curve ..... 45
6.3.5 Internal 8MHz RC oscillator - voltage-frequency characteristic curve ..... 46
6.3.6 Internal 8MHz RC oscillator - Temperature-frequency characteristic curve ..... 46
**7 Package size ..... 47**
7.1 8PIN package size ..... 47
7.2 6PIN package size ..... 49

# 1. General description

## 1.1 Functional features

### CPU configuration

- 1K×14-Bit OTP ROM
- 48×8-Bit SRAM
- 5 levels of stack space
- 6-level programmable Voltage Reset (LVR)
  1.8 V / 2.4 V / 2.7 V / 3.3 V / 3.6 V / 3.9 V
- Operating current less than 2 mA (4MHz/5V)
- When working current is less than 5μA (11 KHZ / 5 v)
- Sleep current less than 1 μA(sleep mode)

### I/O configuration

- 1 set of bidirectional IO ports: P60 to P65
- Low six I/O pins
- Wake up port: P6
- 6 programmable pull-up I/O pins
- 6 programmable pull-down I/O pins
- P63(reset pin) configurable pull-down and output
- In the low external fracture: P60

### Working voltage, temperature

$V_{LVR2.7V}$ ~ 5.5 V | Fcpu = 0 ~ 8 MHZ

$V_{LVR2.4V}$ ~ 5.5V|Fcpu=0 ~ 4MHz

$V_{LVR1.8V}$ ~ 5.5V|Fcpu=0 ~ 2MHz

Low operating temperature range:

Working temperature -40℃-85℃

### Working frequency range

- Built-in IHRC oscillation circuit:
  8MHz/1MHz/910KHz
- Built-in ILRC oscillation circuit:
  11KHz
- Low command clock frequency division to choose:
  2Clock, 4Clock, 8Clock，16 clock, 32 clock

### Peripheral modules

- 8Bit real-time clock/counter
- 12Bit pulse width modulator PWM

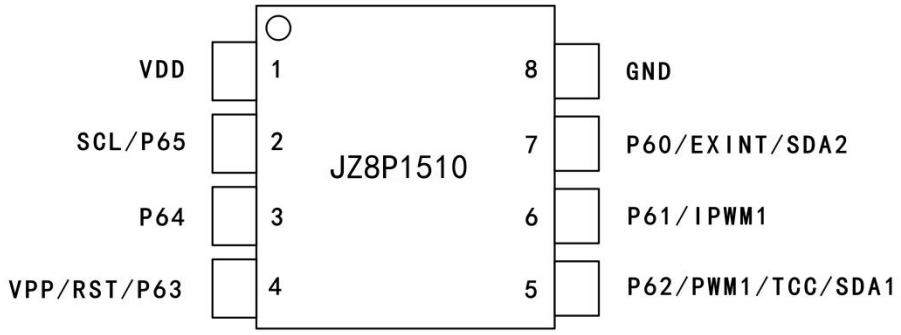
### Interrupt source

- TCC overflow interrupt
- External interrupt
- Input port state changes to generate interrupts
- T1/PWM cycle overflow interrupt

### Type of package

- JZ8P1510-DIP8
- JZ8P1510-SOP8
- JZ8P1510-SOT23-6

## 1.2 Pin assignment




**JZ8P1510-8 pin map**

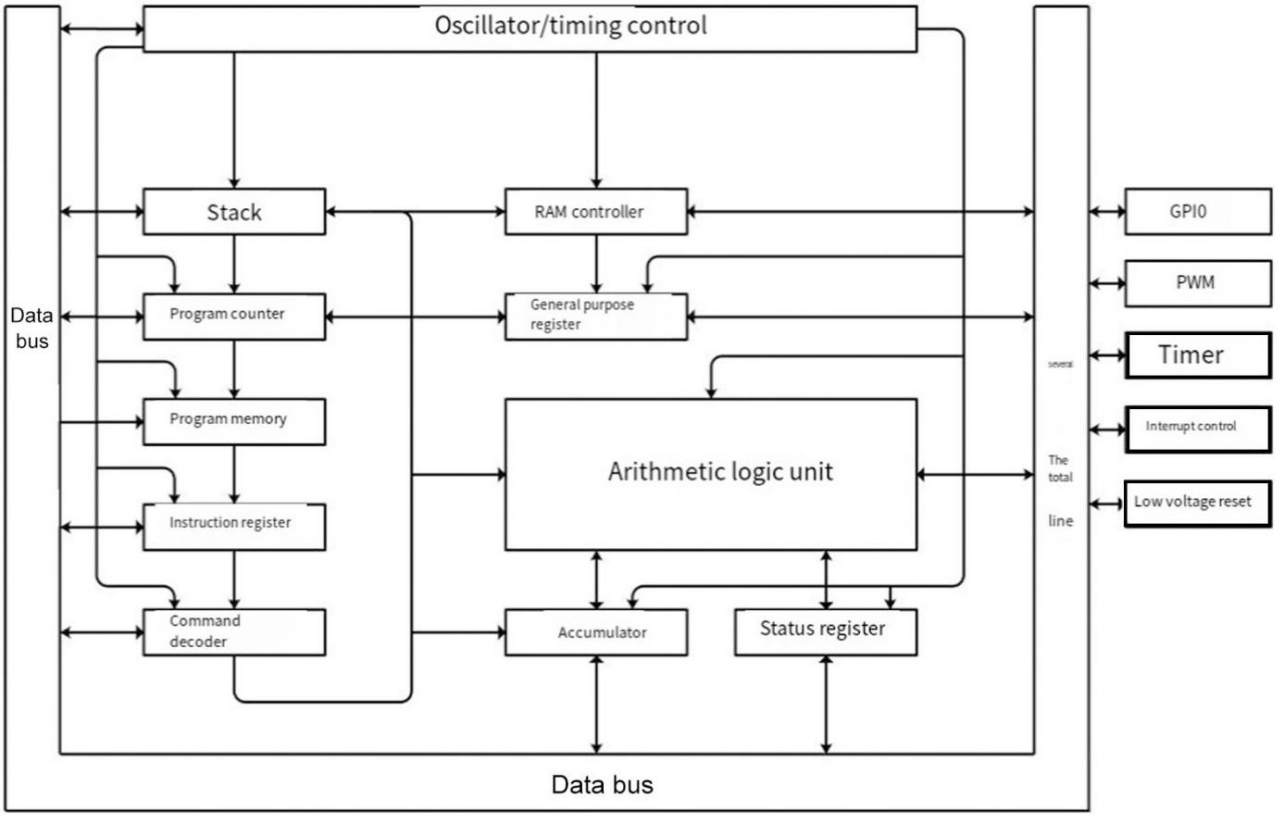



**JZ8P1510-6 pin map**

## 1.3 Pin description

| Serial number | Pin name | I/O | Functional Description |
|---|---|---|---|
| P60 | P60 | I/O (up/down) | GPIO, programmable pull up and down, high drive, port wake-up |
| | EXINT | I (SMT) | External interrupt input port |
| | SDA2 | I (SMT) | Burn data port |
| P61 | P61 | I/O (up/down) | GPIO, programmable pull up and down, high drive, port wake-up |
| | IPWM1 | 0 | PWM1 complementary (reverse) outlet |
| P62 | P62 | I/O (up/down) | GPIO, programmable pull up and down, high drive, port wake-up |
| | TCC | I | External TCC source input foot |
| | PWM1 | 0 | PWM1 outlet |
| | SDA1 | I (SMT) | Burn data port |
| P63 | P63 | I/O (pull up) | GPIO, programmable pull-up, port wake-up |
| | RST | I (SMT) | Reset pin |
| | VPP | I | Burn the high pressure port |
| P64 | P64 | I/O (up/down) | GPIO, programmable pull up and down, high drive, port wake-up |
| P65 | P65 | I/O (pull up/pull down) | GPIO, programmable pull up and down, high drive, port wake-up |
| | SCL | I (SMT) | Burn the clock port |
| | VDD | -- | Power supply |
| | VSS | -- | to |

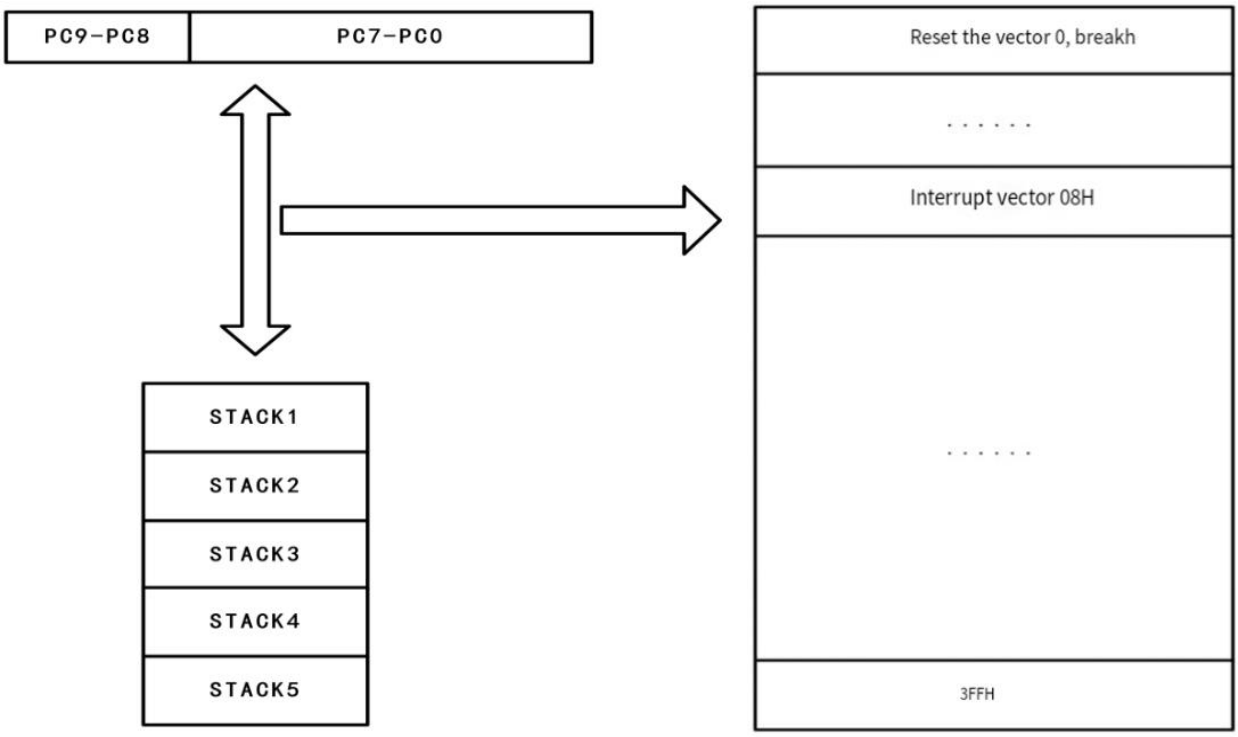
## 1.4 System block diagram




**System circuit block diagram**

# 2. Memory structure

## 2.1 Program storage area




Program store structure diagram

## 2.2 The data store

**R \ IOC pages register area (IOC for specific pages, can only use the IR/IW instruction in reading and writing operations)**

| Address | R page register | IOC page register |
|---|---|---|
| 0x00 | R0/IAR(Indirect addressing register) | keep |
| 0x01 | R1 / TCC (Timing counter) | Hold on |
| 0x02 | R2/PC(Program counter) | CONT(Control register) |
| 0x03 | R3/STATUS(Status register) | Keep |
| 0x04 | R4/RSR(RAM select register) | Reserve |
| 0x05 | Reserved | Keep |
| 0x06 | R6/P0RT6(P6 data register) | IOC6/P6CR(P6 direction control register) |
| 0x07 | Reserved | IOC7/T1L (T1/PWM counter low) |
| 0x08 | R8/PWMCON(PWM control register) | IOC8/T1H (T1/PWM counter high bit) |
| 0x09 | R9/PRDL(PWM periodic low bit register) | IOC9 / PDCR0 (Port drop-down control register 0) |
| 0x0A | RA/PDCL(PWM duty cycle low bit register) | keep |
| 0x0B | RB/PRDCH(PWM cycle duty cycle high bit register) | IOCB/PDCR1 (Port drop-down control register 1) |
| 0x0C | Hold on | keep |
| 0x0D | RD/ICIECR(Input state change interrupt enabled) | IOCD/PHCR(Port pull-up control register) |
| 0x0E | RE/CPUCON (CPU mode control register) | IOCE/EISCR (EIS control register) |
| 0x0F | RF/ISR(Interrupt flag register) | IOCF/IMR(Interrupt Enable control register) |
| 0x10<br>…<br>0x3F | General purpose registers | |

# 3 Functional description

## 3.1 Operate registers

### 3.1.1 RPAGE ~ R0/IAR (Indirect addressing register)

| 00H(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **IAR** | RIND<7:0> | | | | | | | |
| **Read/Write** | R | | | | | | | |
| **Reset value** | X | | | | | | | |

The indirection addressing register is not an actual register; its main function is to act as a pointer to indirection addressing. Any instructions with R0 as Pointers, R4 is actual corresponding address (RAM select register) low six RSR < 5-0 > data points.

### 3.1.2 RPAGE ~ R1 / TCC (Timing counter)

| 01H(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **TCC(R)** | TCC<7> | TCC<6> | TCC<5> | TCC<4> | TCC<3> | TCC<2> | TCC<1> | TCC<0> |
| **Read/Write** | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TCC is an 8Bit uplink counter, clock source optional internal clock/external clock, count overflow can form interrupts, TCC can be read and written.

TCC can be triggered by a signal edge on the EXINT pin or by a command cycle to generate a plus 1 operation (defined by the CONT.4 bit). If PAB bit (CONT.3) is cleared, a prescaler is assigned to TCC, and the prescaler value is cleared when a value is written to the TCC register.

### 3.1.3 RPAGE ~ R2/PC(Program counter)

| 02H(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **PC** | PC<7> | PC<6> | PC<5> | PC<4> | PC<3> | PC<2> | PC<1> | PC<0> |
| **Read/Write** | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A program counter (PC) is a pointer used to record the instructions to be processed by the CPU in each instruction cycle. During the CPU run cycle, the PC pushes the instruction pointer into the program memory, and the pointer is incremented by 1 to enter the next cycle. The JZ8P1510 has a 10-bit wide program counter (PC) whose low bytes come from the read-write PCL and high bytes (PC<9:8>) are unreadable.

Stack is used to record the instruction pointer returned by the program. When a subprogram is called, the PC pushes the instruction pointer onto the stack. When the return instruction is to be executed, the stack sends the instruction pointer back to the PC to continue the original process. JZ8P1510 has 5 levels stack, neither program storage space or the stack of data storage, and the stack pointer can't read and write.

(1) Both the register PC and the built-in 5-level stack are 10 bits wide, used for addressing 1K×14Bit ROM, and the JZ8P1510 program memory area mapping.

(2) In general, the PC is incremented by one; When reset, all bits of PC are reset.

(3) Instruction "JMP" allows direct loading of the low 10 bit address, therefore, JMP instruction can achieve the current page (within the range of 1K) any position jump. Instruction "JMP" directly loads the low 10 bit address, and pushes PC +1 onto the stack, so that the subroutine entry address can be accurately located as long as it is in the same page.

(4) When executing "RET" instruction, the top data of the stack is sent to the PC.

(5) When setting the table lookup ability of PC to 1/4K (setting the table lookup range from 0 to 256 in the OPTION option), any instruction that directly modifies the content of PC will cause the 9th and 10th bit of PC to be reset. Therefore, the jump generated is limited to the first 256 addresses of the same page, and the instruction to change the content of the PC needs 2 instruction cycles.

When the lookup ability of PC is set to 1K, any instruction that rewrites the value of PC will affect the top two bits of PC accordingly. Therefore, the generated jumps can be expanded to the 1K range.

(6) When an interrupt occurs, the value of the program counter changes, and PC is assigned the value 0x08.

(7) The stack works like a loop buffer, that is, after five stack pushes, the data in the sixth push overwrites the data in the first push, the data in the seventh push overwrites the data in the second push, and so on.

### 3.1.4 RPAGE ~ R3/STATUS(status register)

| 03H(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **STATUS** | RST | GB1 | GB0 | T | P | Z | DC | C |
| **Read/Writ** | R | R/W | R/W | R | R | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 1 | 1 | X | X | X |

Bit<7> : Rst-reset type flag bit

0: Other reset type

1: If the sleep mode is awakened by changing the pin state

Bit< 6:5 > : GB1: GB0- Undefined, general-purpose read/write bit

Bit<4> : T- time overflow bit

0: Other

1: Perform the "SLEEP" command or low pressure reset

Bit<3> : P- power down flag bit

0: Execute the "SLEEP" command

1: Power on and reset

The events that affect T/P are shown in the following table:

| Type | RST | T | P |
|---|---|---|---|
| Power-on reset | 0 | 1 | 1 |
| Press RESET in working mode | 0 | Keep | Keep |
| RESET Wake up | 0 | 1 | 0 |
| Port state change wake up | 1 | 1 | 0 |
| Execute the SLEEP command | Keep | 1 | 0 |

Bit<2> : z-zero flag bit set to "1" when the result of an arithmetic or logical operation is zero

0: When the result of an arithmetic or logical operation is not 0

1: When the result of an arithmetic or logical operation is 0

Bit<1> : DC- auxiliary carry flag

0: When performing the addition, no carry is generated for the lower four bits;

When performing a subtraction operation, the lower four bits produce a borrow

1: When performing the addition operation, the lower four bits have carry generation;

When performing subtraction, no borrow is generated for the lower four bits

Bit<0> : C- Carry flag

0: When performing the addition, no carry is generated for the upper four bits; / when performing subtraction, the high four bits produce a borrow

1: When performing the addition operation, the higher four bits have carry generation; / When performing subtraction, no borrows are generated

### 3.1.5 RPAGE ~ R4/RSR(RAM select register)

| 04H(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **RSR** | - | - | RSR<5> | RSR<4> | RSR<3> | RSR<2> | RSR<1> | RSR<0> |
| **Read/Write** | R | R | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 1 | 1 | X | X | X | X | X | X |

RSR<5:0> Used to select RAM register address in indirect addressing mode (addressing range: 0X10 to 0X3F)

RSR is used to cooperate with R0 to realize indirect addressing operation. You can place the address corresponding to a register in the RSR and then indirectly address register R0 by accessing it, at which point the address points to the register at the corresponding address in the RSR.

### 3.1.6RPAGE ~ R6/P0RT6(P6 data register)

| 06H(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **PORT6** | - | - | P65 | P64 | P63 | P62 | P61 | P60 |
| **Read/Write** | R | R | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Port input/output register, 6 bits for P6 port

### 3.1.7 RPAGE ~ R8/PWMCON(PWM control register)

| 08H(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **T1** | T1EN | BZEN | IPWMEN | PWMEN | T1PTEN | T1PSR<2> | T1PSR<1> | T1PSR<0> |
| **Read/writ** | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit<7> : T1EN-T1 /PWM counter enable control

1: Enable

0: Forbid

Bit<6> : BZEN-BUZZER output enable control bit

1: Enable BUZZER output function, T1 overflow BUZZER flip (PWMEN=0)

0: Forbidden

Bit<5> : IPWMEN-IPWM enables the control bit

1: Enable PWM complementary output (PWM invert)

0: Disable

Bit<4> : PWMEN-PWM enables the control bit

1: Enable PWM output (BZEN=0)

0: Disable

Bit<3:0> : T1PTEN, T1PSR2 ~ T1PSR1 frequency division coefficient selection bit:

| T1PTEN | T1PSR<2> | T1PSR<1> | T1PSR<0> | Frequency division ratio |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1:1 |
| 1 | 0 | 0 | 0 | 1:2 |
| 1 | 0 | 0 | 1 | 1:4 |
| 1 | 0 | 1 | 0 | 1:8 |

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1:16 |
| 1 | 1 | 0 | 0 | 1:32 |
| 1 | 1 | 0 | 1 | 1:64 |
| 1 | 1 | 1 | 0 | 1:128 |
| 1 | 1 | 1 | 1 | 1:256 |

### 3.1.8 RPAGE ~ R9/PRDL(PWM periodic low bit register)

| 09H(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **PRDL** | PRD<7> | PRD<6> | PRD<5> | PRD<4> | PRD<3> | PRD<2> | PRD<1> | PRD<0> |
| **Read/Write** | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit<7:0> : PRD<7:0>-PWM cycle low octet data

### 3.1.9 RPAGE ~ RA/PDCL(PWM duty cycle low bit register)

| 0AH(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **PDCL** | PDC<7> | PDC<6> | PDC<5> | PDC<4> | PDC<3> | PDC<2> | PDC<1> | PDC<0> |
| **Read/Write** | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit<7:0> : PDC<7:0>-PWM duty cycle lower eight bits of data

### 3.1.10 RPAGE ~ RB/PRDCH(PWM Cycle Duty Cycle High bit register)

| 0BH(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **PRDCH** | PRD<11> | PRD<10> | PRD<9> | PRD<8> | PDC<11> | PDC<10> | PDC<9> | PDC<8> |
| **Read/Write** | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit<7:4> : PRD<11:8>-PWM cycle high four bits of data

Bit<3:0> : PDC<11:8>-PWM duty cycle is four bits higher

### 3.1.12 RPAGE ~ RD/ICIECR(Input state change interrupt enabled)

| 0DH(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **ICIECR** | - | - | IEN<5> | IEN<4> | IEN<3> | IEN<2> | IEN<1> | IEN<0> |
| **Read/Write** | R | R | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit<5:0> : IEN<5:0>-P6 Input state change interrupt enable control bit

1: Enable

0: Forbid

Note: Only works if the OPTION option [P6 port wake up] is [independent control (register 0x0D)].

### 3.1.13 RPAGE ~ RE/CPUCON(CPU mode control register)

| 0EH(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **CPUCON** | IPWM | PWMCKS | TCCCKS | PWMWE | TCCWE | STPHX | CLKMD | IDLE |
| **Read/Write** | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit<7> : IPWM-PWM complementary (negated) output enable bit

1: Enable complementary output

0: Disable complementary output

Bit<6>：PWMCKS-PWM Clock source selection bit

1: Select the system clock

0: Select the instruction cycle clock

Bit<5>：TCCCKS-TCC Clock source selection bit

1: choose the system clock/low speed oscillator clock

0: Select instruction cycle clock/external clock

Bit<4>：PWMWE-PWM Wake up enable bit

1: Enable PWM to wake up, can wake up the idle mode

0: PWM sensei is prohibited

Bit<3>：TCCWE-TCC Wake up enable bit

1: Enable TCC to wake up, can wake up the idle mode

0: Disable TCC wakeup

Bit<2>：STPHX-High speed clock control bit

1: Stop the high speed clock

0: The high-speed clock is working properly

Bit<1>：CLKMD- Clock source selection bit

1: System clock using low speed RC oscillator clock

0: System clock using high speed RC oscillator clock

When the system enters the low speed mode from the high speed mode, set CLKMD=1 first, then set STPHX=1;

When the system enters the high speed mode from low speed mode, set STPHX=0 first, and then set CLKMD=0.

Bit<0>：IDLE-Idle mode select bit

1: The system enters idle mode when it executes the SLEEP command, and the system clock works normally

TCC and PWM continue to work in idle mode if the system clock is selected, and can wake up the system..

0: The system goes into SLEEP mode when it executes the sleep command

### 3.1.14 RPAGE ~ RF/ISR(Interrupt flag register)

| 0FH(R) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **ISR** | - | - | - | - | T1IF | EXIF | ICIF | TCIF |
| **Read/Write** | R | R | R | R | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit<7:4> : Unused

Bit<3> : T1IF-T1/PWM cycle interrupt flag

Bit<2> : EXIF- external port interrupt flag bit

Bit<1> : ICIF-P6 port state change interrupt flag bit

Bit<0> : TCIF-TCC interrupt flag bit

1: There is an interrupt, software clear 0

0: no interruption

Note: when you clear the interrupt flag bit, must use the MOV RF, A operation, cannot use BTC and AND RF, A command operation.

## 3.2 Control registers

### 3.2.1 CONT(Control register)

| 01H(IOC) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **CONT** | LRCEN | INT | TS | TE | PAB | PSR2 | PSR1 | PSR0 |
| **Read/Write** | R/W | R | R/W | R/W | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Bit<7>：LRCEN-Internal low speed RC oscillator enable bit

1: Enable

0: Forbid

Bit<6> : INT- Interrupt enable flag bit

0: Interrupt forbidden by instruction or hardware

1: Interrupt enabled by instruction

Bit<5> : TS-TCC signal source selection bit

0: Internal instruction cycle clock (TCCCKS=0)/system clock (TCCCKS=1)

1: External input clock (TCCCKS=0)/low speed oscillator clock (TCCCKS=1)

Bit<4>：TE-TCC Signal edge selection bit

0: TCC pin signal changes from low to high plus 1

1: TCC pin signal change from high to low add 1

Bit<3>：PAB-Pre-divider allocates bits

0: predivider is assigned to TCC

1: Unused

Bit<2:0> : PSR2 ~ PSR0-TCC pre-frequency selection control bit:

| PSR2 | PSR1 | PSR0 | TCC frequency division coefficient |
|---|---|---|---|
| 0 | 0 | 0 | 1:2 |
| 0 | 0 | 1 | 1:4 |
| 0 | 1 | 0 | 1:8 |
| 0 | 1 | 1 | 1:16 |
| 1 | 0 | 0 | 1:32 |
| 1 | 0 | 1 | 1:64 |
| 1 | 1 | 0 | 1:128 |
| 1 | 1 | 1 | 1:256 |

### 3.2.2 IOPAGE ~ IOC6/P6CR(P6 direction control register)

| 06H(IOC) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| P6CR | - | - | P6CR<5> | P6CR<4> | P6CR<3> | P6CR<2> | P6CR<1> | P6CR<0> |
| Read/write | R | R | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Port6 direction control bit

1: Input

0: Output

### 3.2.2 IOPAGE ~ IOC7/T1L(T1/PWM counter low bit)

| 07H(IOC) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| T1L | T1<7> | T1<6> | T1<5> | T1<4> | T1<3> | T1<2> | T1<1> | T1<0> |
| Read/write | R | R | R | R | R | R | R | R |
| Reset value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

T1/PWM low octet count value (read only)

### 3.2.2 IOPAGE ~ IOC8/T1H(T1/PWM counter high bit)

| 08H(IOC) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| T1H | - | - | - | - | T1<11> | T1<10> | T1<9> | T1<8> |
| Read/write | R | R | R | R | R | R | R | R |
| Reset value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

T1/PWM high four-bit count value (read only)

### 3.2.3 IOPAGE ~ IOC9/PDCR0(port pull-down control register 0)

| 09H(IOC) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| PDCR0 | - | - | P6PD<5> | P6PD<4> | - | - | - | - |
| Read/write | R | R | R/W | R/W | R | R | R | R |
| Reset value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bit<5:4> : P6<5:4> pull-down enable control

0: Enable

1: Forbid

Note: when the OPTION options 】 【 P6 drop-down control can make 】 【 IOC9 / PHDCR register operational software.

### 3.2.4 IOPAGE ~ IOCB/PDCR1(Port pull-down control register 1)

| 0BH(IOC) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| PDCR1 | P6PD<3> | P6PD<2> | P6PD<1> | P6PD<0> | - | - | - | - |
| Read/write | R/W | R/W | R/W | R/W | R | R | R | R |
| Reset value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Bit<7:4> : P6<3:0> pull-down enable control

0: Enable

1: Forbid

Bit<3:0> : Undefined, fixed to 1

### 3.2.5 IOPAGE ~ IOCD/PHCR(Port pull-up control register)

| 0DH(IOC) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| PHCR | - | - | P6PH<5> | P6PH<4> | P6PH<3> | P6PH<2> | P6PH<1> | P6PH<0> |
| Read/write | R | R | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Port6 pull-up control

0: Enable

1: Forbid

Note: When the OPTION option [reset port pull up] selects [Disable] and [P6 pull up control] selects [Enable], the P6PH<3> software can be operated. If [reset port pull up] selects [Enable], the P63 pull up function is enabled and cannot be changed by software.

### 3.2.6 IOPAGE ~ IOCE/EISCR(EIS Control register)

| 0EH(IOC) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| EISCR | - | EIS | - | - | - | - | - | - |
| Read/Write | R | R/W | R | R | R | R | R | R |
| Reset value | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |

Bit<7>： Not In Use

Bit<6> : EIS-P60 external interrupt enable bit

1: can make, in this case, P60 of I/O control (P6CR Bit0) must be set to "1", the state of the pin can be read by the P6 port, falling edge P60 trigger external interrupt

0: Forbidden, P60 is a two-way I/O pin, and the EXINT channel is blocked

Bit<5:0> : Not used

### 3.2.7 IOPAGF ~ IOCF/IMR(Interrupt enable control register)

| 0FH(IOC) | Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
|---|---|---|---|---|---|---|---|---|
| **IMR** | - | - | - | - | T1IE | EXIE | ICIE | TCIE |
| **Read/Write** | R | R | R | R | R/W | R/W | R/W | R/W |
| **Reset value** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit<7:4> : Unused

Bit<3> : T1IE-T1 interrupt enable control

1: Enable

0: Forbid

Bit < 2 > : EXIE - the external interrupt control

1: can make falling edge (P60 trigger external interrupt)

0: ban

Bit < 1 > : ICIE - P6 port state change the interrupt control

1: can make

0: Forbid

Bit < 0 > : TCIE - TCC can make control overflow interrupt

1: Enable

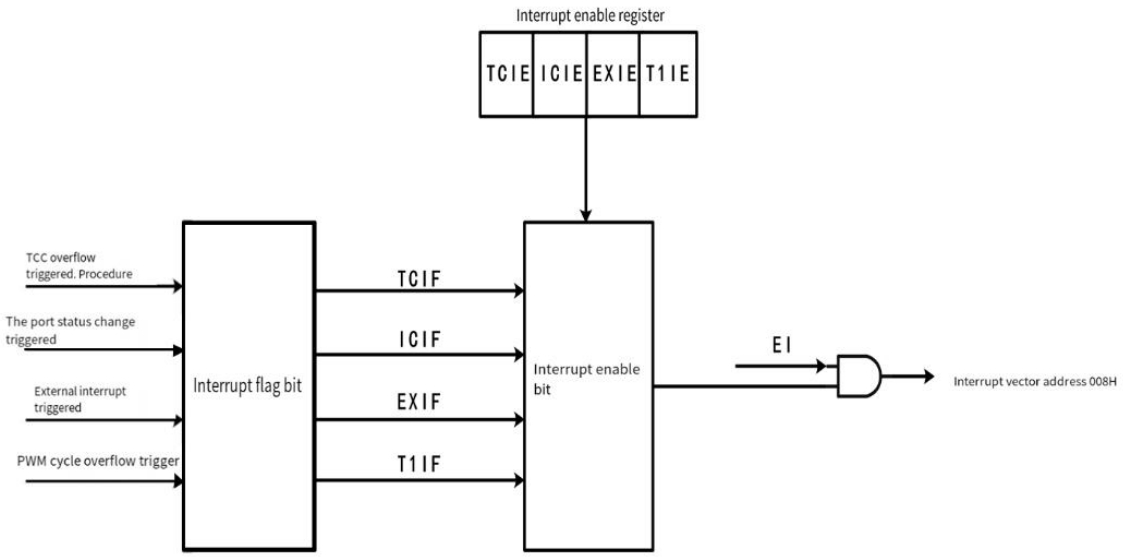
0: Forbid

## 3.3 Interrupt

JZ8P1510 has four interrupt source, whether using that an interrupt, must make can always interrupt, namely "EI" instruction. Interrupt vector for fixed default address 008 h. The following are the characteristics of each interrupt:

| Interrupt type | Interrupt source | Enabling condition | Interrupt flag |
|---|---|---|---|
| Exterior | External interruption | EI + EXIE=1 | EXIF |
| Outside | Port input change interrupt | EI + ICIE=1 | ICIF |
| Inside | TCC overflow interrupted | EI + TCIE=1 | TCIF |
| Inside | PWM cycle overflow interrupt | EI + T1IE=1 | T1IF |

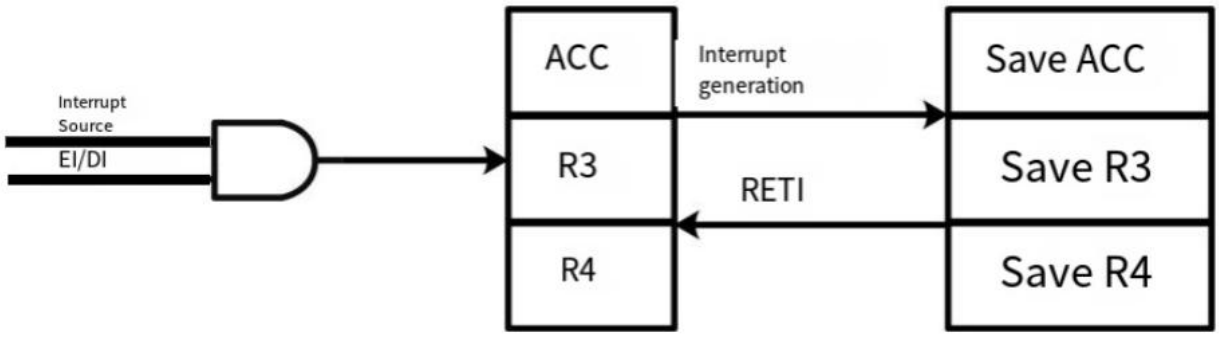
RPAGE's RF is the interrupt status flag register, which records the interrupt flag bit after an interrupt request is generated. IOCF for interrupt register interrupt the allowed and forbidden in the two register Settings. Always interrupt is allowed by the "EI" instruction, on the contrary, always interrupt is prohibited by the "DI" instruction. When an interrupt occurs, its execution of the next instruction will be executed from the interrupt vector address 008 h. Before leaving the interrupt service routine the corresponding interrupt flags must be reset, so as to avoid misoperation interrupt.




The interrupt principle diagram

### 3.3.1 Interrupt site protection

In response to a disruption in the process, it is strongly recommended that in the program using the interrupt protection, save the contents of ACC, R3 and R4, until leave the interrupt subroutine, then reload the ACC will be saved value, R3 and R4, so to avoid when the interrupt subroutine, instruction will be ACC, R3 and R4 value change, An error occurred while trying to return to the main program. As shown in the figure below:




Schematic diagram of software interrupt site protection

## 3.4 Reset

### 3.4.1 Overview of reset function

The JZ8P1510 system provides 3 reset methods:

- POR electric reset
- RESET input low level RESET foot
- As low voltage reset

More than any other type of reset occurs, the initialization register all of the system to reset value, stop running program, the program counter (PC) reset at the same time. After the reset, the system starts running again from the vector 0000H.

Any kind of reset condition requires a certain amount of response time, system reset mechanism can ensure the reliable reset of the MCU. Different types of oscillator, the time required to complete restoration is also different. Therefore, VDD rising velocity and different oscillator start up time is not fixed. RC oscillator initial time for the shortest, crystal oscillator start up time is longer. Should be paid attention to in the process of user terminal use, consider scenarios requirement for electric reset time.

### 3.4.2 POR electric reset

Power-on reset is closely related to LVR operation. The power-on process of the system is in the form of a gradually rising curve, and it takes a certain time to reach the normal level value.

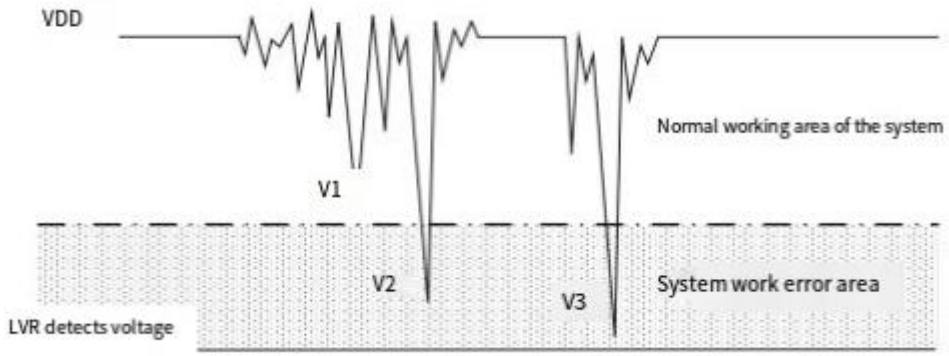
- Power on: the system detects the rise of the power supply voltage and waits for it to stabilize;
- External reset (limited to external reset pin enabled state) : The system detects the external reset pin state. If it is not high, the system remains reset until the external reset pin is released;
- System initialization: all system registers are set to their initial values;
- Oscillator starts working: the oscillator starts to supply the system clock;
- Execute the program: the power on ends, and the program starts to run;

Electric reset time is determined by the OPTION of select reset time 】【, shown in the table below:

| PWRT | Reset setup time |
|---|---|
| PWRT | 4.5 ms |
| PWRT | 18ms |
| PWRT | 72ms |
| PWRT | 288ms |
| PWRT | 140us |

### 3.4.3 LVR low voltage reset

Using reset for external factors of system voltage drop situation (for example, the interference or the change of the external load), off electricity may cause the system working status is not normal or program error.




Voltage drop dead zone may enter the system. System dead zone means that the power supply can not meet the requirement of the system, the minimum operating voltage. Pictured above is a typical power reset. In this picture, the VDD is severely disturbed and the voltage drop is very low. Dotted line more than regional system to work properly, within the area below the dotted line, the working state of the system into the unknown, this area is called a dead zone. When the VDD fell to V1, the system is still in the normal condition; When the VDD fell to V2 and V3 system into the dead zone is prone to error. May enter the dead zone of the following systems:

**DC in use:**

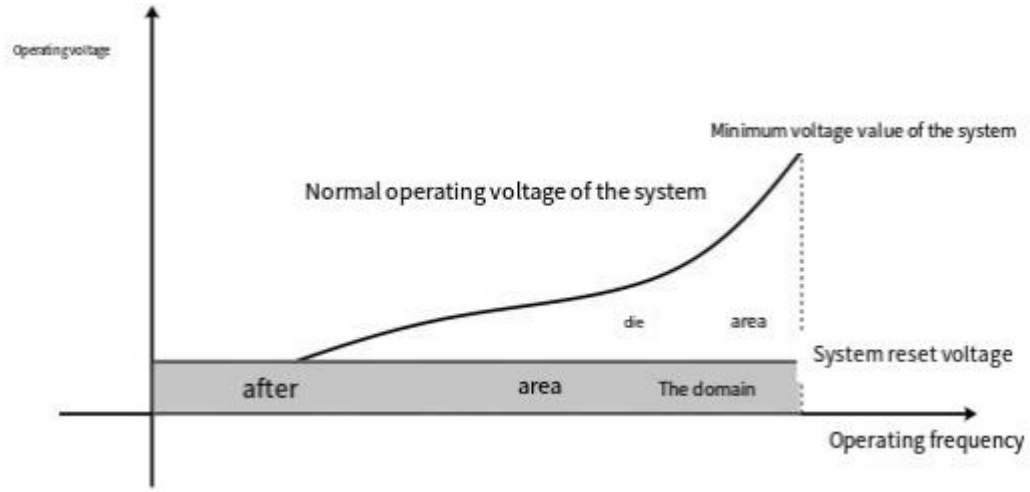
Commonly used in DC using battery power, when the capacity of the battery voltage is too low or the microcontroller drive system voltage may fall and into the dead zone. At this moment, not further down to as testing voltage, so the system at a dead zone.

**In using AC:**

When using the AC power supply system, DC voltage value is influenced by the noise of the AC power supply. When the external load is too high, such as driving motor, load movement interference also affect the DC power supply. If VDD due to interference drop below the minimum operating voltage, then the system is likely to enter the unstable working condition. Used in AC system, the electric time is longer. Among them, the temporal protection system on the normal electricity, but the electrical process and application of DCS in situations similar to AC power off patrol, VDD voltage in the process of slow down easy into the dead zone.

### 3.4.4 Working frequency relationship with as low pressure detection

In order to improve the performance of the system power reset, must first clear system has the lowest operating voltage value. System minimum working voltage and speed of execution, the values of the minimum operating voltage under different speed of execution is different also.




As shown in the above, the normal work of the system voltage area generally higher than the system reset voltage, at the same time reset voltage is determined by the low voltage detection (as) level. When the system execution speed increase, the system also increase the minimum operating voltage, but because the system reset voltage is fixed, therefore the minimum operating voltage in the system and the system reset voltage will appear a voltage between area, system can not work normally, also won't reset, this area is the dead zone.

To avoid voltage dead zone, in choosing a working frequency, to select the corresponding points as reset voltage. The following table:

| IRC Frequency | The system clock frequency doubling | Clocks frequency division | LVR recommends voltage points |
|---|---|---|---|
| IRC-8MHz | Frequency doubling | 2 Clocks | As = 2.7 V |
| IRC-8MHz | Not octave | 2 Clocks | As = 2.4 V |
| IRC-910KHz | Frequency doubling | 2 Clocks | As = 1.8 V |
| IRC-910KHz | No frequency doubling | 2 Clocks | As = 1.8 V |

Note: 1. the working frequency = instruction cycle frequency = IRC + Clocks frequency division;

2. this working frequency and the corresponding value of the LVR reset voltage point, only recommended value, the user in the use of the process, according to the specific application can be adjusted reset voltage point.

## 3.5 Working mode

The JZ8P1510 can operate at different clock frequencies in four operating modes, which can control the operation of the oscillator, the execution of the program, and the functional loss of the analog circuit.

- High speed mode: the system clock selects high speed IRC clock;
- Low speed mode: the system clock choose internal clock at low speed;
- Idle mode: the system clock works normally, and other parts go to sleep (TCC and PWM select the system clock to continue to work and wake up the system, and the port state changes to wake up the system);
- Sleep mode: all functions stop working, the system goes to sleep, and the port state can be woken up;

| Function module | High speed mode | Low Speed mode | Idle mode | Sleep mode |
|---|---|---|---|---|
| IHRC | Run | STPHX control | STPHX controls | stop |
| ILRC | Run | Run | Run | stop |
| CPU instructions | Execution | Execution | Stop | Stop |
| TCC | Workable | Workable | Be available for work | Stop |
| PWM | Workable | Workable | Workable | Stop |
| Internal interruption | All valid | All valid | TCC,PWM valid | TCC(P62) |
| External interruption | Valid | Effective | ICIE | ICIE |
| Wakeup function | - | - | P6IC,TCC,PWM | P6IC |

### 3.5.1 High-speed mode

High-speed mode is the high speed system clock working mode, the system clock source provided by the high-speed RC oscillator. The program is executed. On electric reset or any other type of reset when triggered, the system into high speed mode execute a program. When the system from sleep mode awakened after high speed mode. High speed mode, the normal work of the high speed oscillator, power consumption is the largest.

- The program is executed and all functions can be controlled;
- The system rate is high;

- Both the high speed oscillator and the internal low speed oscillator work normally;
- Through the CPU mode control register, the system can switch from high-speed mode to any other kind of work mode;
- System after wake up from sleep patterns into high-speed mode;
- Low speed mode you can switch to the mode;
- From the model into the idle mode, wake up after the return to the high-speed mode;

### 3.5.2 Low-speed mode

Low speed mode for the system clock work at low speed. The system clock source provided by the internal RC oscillator at low speed. Low speed mode CLKMD position control mode control register by the CPU. When CLKMD = 0, system for high-speed mode; When the CLKMD = 1, system into the low speed mode. Enter the low speed mode, does not automatically banned the oscillator, SPTHX bit must be put through to prohibit to reduce power consumption.

- Program is executed, all functions can be controlled;
- The system rate is low speed;
- The internal low-speed RC oscillator works normally, and the high-speed oscillator is controlled by STPHX=1. In low speed mode, it is highly recommended to stop the high speed oscillator;
- Through the CPU mode control register, the low speed mode can be switched into other operating modes;
- From low speed mode to sleep mode, wake up and return to low speed mode;
- High speed mode can be switched to low speed mode;
- Enter from low speed mode to idle mode, wake up and return to low speed mode;

### 3.5.3 Idle mode

Idle mode is another ideal state. In sleep mode, all function and hardware equipment have been banned, but in idle mode, the system clock keep working, idle mode power consumption is greater than the sleep mode power consumption. In idle mode, the program is not executed, but the timer and PWM with wakeup function are still working normally, and the clock source of the timer and PWM is the system clock that is still working. Idle mode, there are three ways to awaken

system: 1, P6 port state changes trigger; 2, TCC timer wake; 3, PWM wake up. The user can set the timer or PWM for a fixed period, and the system will be woken up when overflow occurs. By CPU IDLE mode control register to decide whether to enter IDLE mode, and when the IDLE = 1, system into IDLE mode.

- Program and all functions are prohibited;
- The timer with wakeup function works properly;
- As the system clock source oscillator to work properly, other oscillator working state depends on the configuration of system working mode;
- Enter from high speed mode to idle mode, return to high speed mode after waking up;
- Enter from low speed mode to idle mode, return to low speed mode after waking up;
- Idle mode of sensei for P6 port state changes trigger sensei, TCC timer overflow and PWM cycle overflow;
- Idle mode TCC and PWM function is still valid;

### 3.5.4 Sleep patterns

Sleep mode is the ideal state of the system, the program is not executed, and the oscillator also stops working. The power consumption of the whole chip is below 1uA. The sleep mode can be triggered by the P6 port state change to wake up. Enter the sleep mode from high speed mode or low speed mode, and return to the corresponding mode after being woken up. The IDLE bit in the CPU mode control register controls whether to enter sleep mode. When IDLE=0, the system enters sleep mode.

- The program stops executing, and all functionality is disabled.
- All of the oscillator, including external, internal oscillator with high speed and low speed high speed oscillator oscillator are stop work;
- Power consumption is lower than 1 ua;
- Enter from high speed mode to sleep mode, and return to high speed mode after waking up;
- By low speed mode into sleep mode, awakened after return to the idle mode;
- The sleep patterns of sensei source for P6 port state changes trigger;

## 3.6 System clock

JZ8P1510 internal integration of the two oscillators, can through the corresponding configuration OPTION to realize. See the table below:

| Oscillator type | instructions |
|---|---|
| IRC (built-in RC oscillator) | Can choose by RCM 910 k/m / 1 to 8 m |

### 3.6.1 Internal RC oscillator

JZ8P1510 provide internal RC models, the default value is 8 MHZ frequency.

Internal RC oscillation pattern includes 8 MHZ, 1 MHZ and 910 KHZ frequency values. By setting the OPTION of configuration, can choose the IRC working frequency, below is their corresponding relationship:

| Firc | IRC frequency |
|---|---|
| 8 M | The IRC frequency is chosen to be 8MHz |
| 1 M | The IRC frequency is chosen to be 1MHz |
| 910K | The IRC frequency is chosen to be 910KHz |

Dividing JZ8P1510 offers a variety of choice, can in the OPTION to choose, suitable for more occasions. The following table:

| Clocks | Clocks frequency division |
|---|---|
| 2clock | Frequency division is 2clock |
| 4clock | The frequency division is 4clock |
| 8clock | The frequency division is 8clock |
| 16clock | Frequency division for 16 clock |
| 32clock | Frequency division is 32clock |

### 3.6.3 Clock module application instructions

- External conditions are different, there may be a slight oscillation modes of the clock frequency difference, when using, should choose according to need reasonable;

## 3.7 I/O port

JZ8P1510 have 1 set of bidirectional I/O port, a total of six input, 6 output, most of the I/O can reuse for other functions.

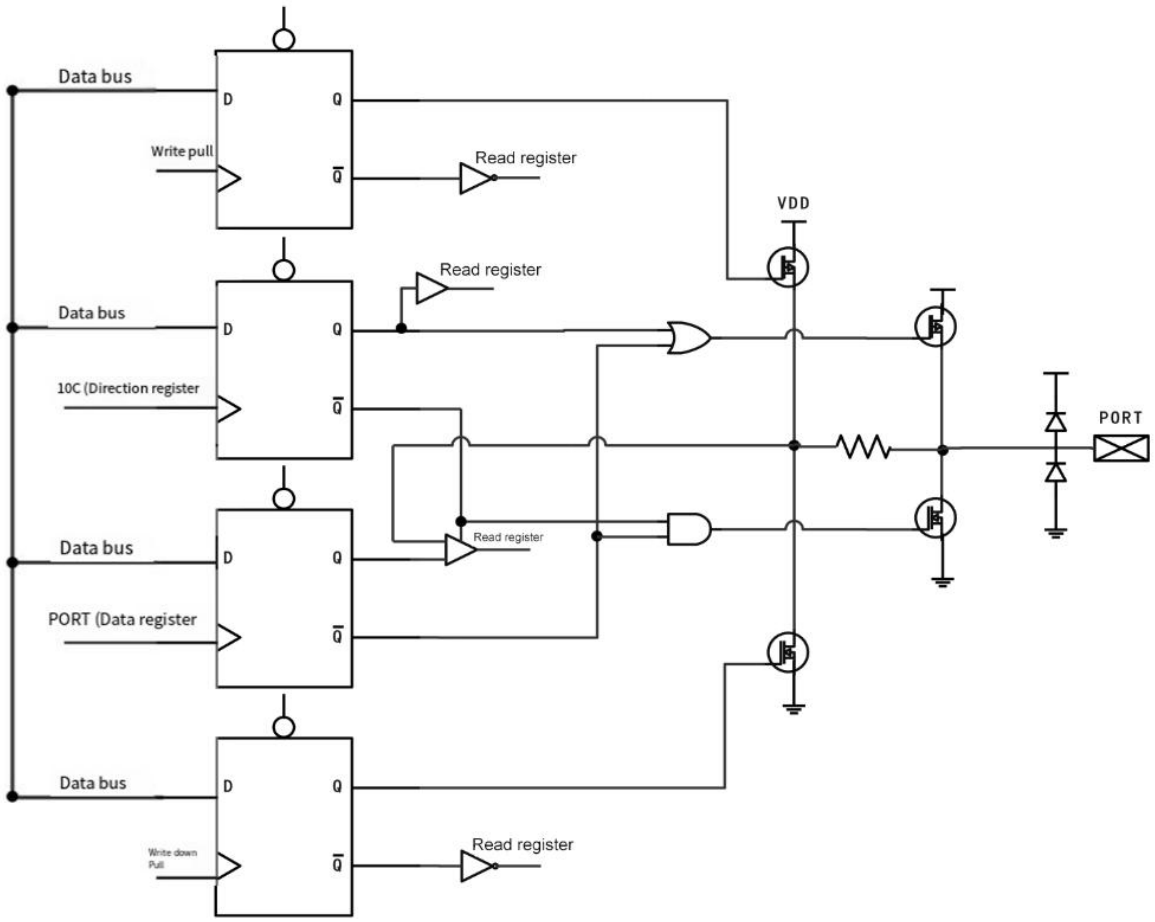
6 on the programmable I/O pins: P60 ~ P65;

Six programmable I/O pins: drop-down P60 ~ P65;

Six programmable wake up I/O pins port state change: P60 ~ P65;

### 3.7.1 GPIO internal structure

The following chart is for reference only understand internal, does not represent the actual circuit.




IO control register data register/ra/dropdown on the circuit structure

### 3.7.2 Port state changes

JZ8P1510 contains six programmable wake up I/O port state change: P60 ~ P65. "SLEEP" instructions that are executed chip can enter the SLEEP mode. At this point, the system clock to stop, all modules to stop working. Port state change sensei can choose to continue the original process by the program (not before SLEEP can make ICIE) or to perform the corresponding jump

(SLEEP before execution EI), and open the corresponding can make control bits, jump to the interrupt subroutine.

**Wake up set port state change query manner**

1, port PORT6 awaken as input;

2, can choose according to need to wake up on the inside of pull or next;

3, enabling port state change interrupt and awaken control independently;

4, to execute commands "SLEEP", SLEEP SLEEP mode;

5, after wake up, the implementation of SLEEP the next instruction;

**Port state change interrupt way sensei Settings**

1. PORT6 port wakeup port is set as input;

2, can choose according to need to awaken the internal pull up and down;

3, enabling port state change interrupt and awaken control independently;

4, can make the PORT PORT state change interrupt;

5, perform "EI" instruction, waiting to enter the interrupt address mouth;

6, under the instruction "SLEEP", SLEEP SLEEP mode;

7. After waking up, it will enter the interrupt address port, exit the interrupt, and execute the next instruction of SLEEP;

### 3.7.3 Port parameters

JZ8P1510 schmidt characteristics of port, form the following (for reference only) :

| port | SMT |
|---|---|
| P6.3 | 0.512 * VDD |
| P6.0 ~ P6.2 | VDD / 0.24 * 0.52 * VDD |
| P6.4 ~ P6.5 | VDD / 0.24 * 0.52 * VDD |

**The above parameters is only for your reference, please refer to our target prototype measured data.**

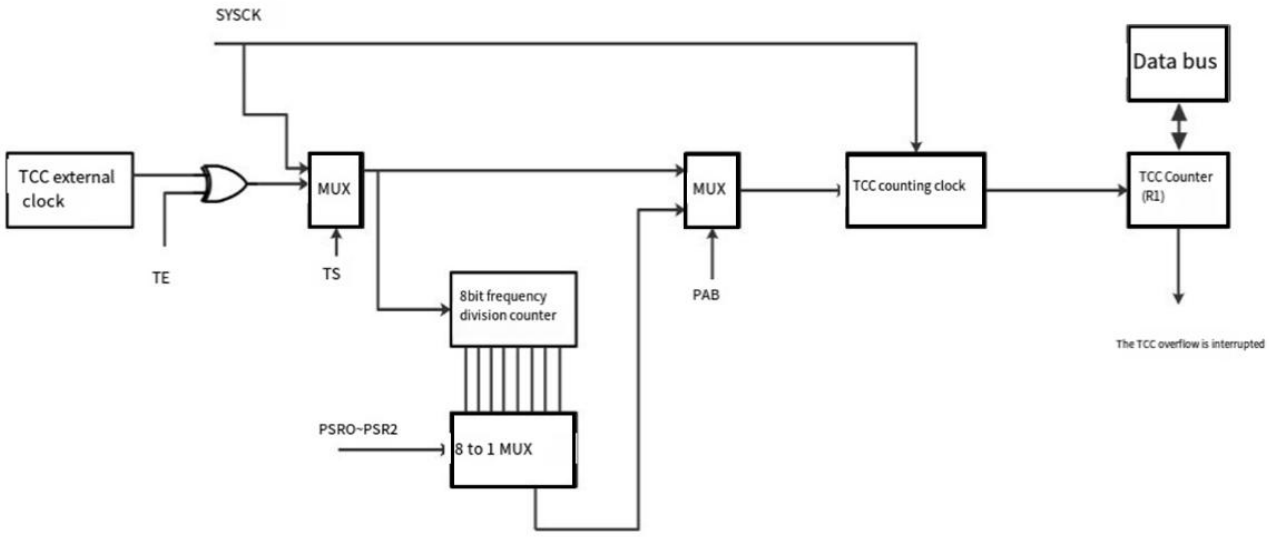
## 3.8 TCC timer counter

JZ8P1510 provide an 8-bit counter as TCC preassigned frequency. Preassigned frequency counter at the same time is only for TCC or unused, and determined by the PAB in the CONT register a preassigned frequency distribution. Three PSR0 ~ PSR2 decided preassigned frequency ratio. Every time in TCC mode TCC is written to a value, frequency preassigned registers are reset.

TCC (R1) is a 8 bit upward counter, as long as there is the clock work. Clock source can be either the internal system clock (rising along the trigger), also can choose external clock (triggered by TCC pin input, the optional), if there is no frequency control, per clock cycle counter plus 1. System to provide a 8 bit counter as TCC preassigned frequency. TCC can be set by CONT register preassigned frequency, triggering along, clock, etc.

TCC counter overflow interrupt signal can be formed.

In IDLE mode, TCC interrupts can awaken circuit, can choose after awakening into interrupt or continue to perform the original procedure, method reference port status changes.




TCC structure diagram

### 3.8.1 TCC timing settings

- Assigning an initial value to the TCC register
- Set the value of the CONT register (choose as a timer or counter and predivision ratio);
- To use as a counter, you need to select the TCC external signal in the CONT register for the positive or negative edge plus 1;

- If you need to perform the interrupt function, you must set the IOCF TCIE bit in a register (Bit0) is 1, and perform the EI instructions;
- Interrupt program part to save manual ACC, the STATUS and R4 in stack machine, RETI instructions executed after, again from the stack, exit the interrupt to clear before TCC interrupt flag bit;

### 3.8.2 TCC showed timing calculation

TCC timing function by write values to TCC register, give it an initial value of timer, timer from the initial value position began to accumulate, until the timer interrupt overflow.

**TCC formula of periodic time clock) select the internal instruction cycle:**

$$\text{TCC timingtime} = (\text{System clock frequency division}/\text{Fosc}) \times (\text{TCC Frequency division}) \times (256 - \text{TCC Initial value})$$

Example:

System clock division = 2clock, Fosc=8 MHz, TCC division selection =4 division, TCC initial value =156;

$$\text{TCC timingtime} = (2/8) \times (4) \times (256 - 156) = 100\ \text{us}$$

**TCC timing time calculation formula (select external input clock):**

$$\text{TCC timingtime} = (\text{External input clock}) \times (\text{TCC Frequency division}) \times (256 - \text{TCC Initial value})$$

Example:

External input clock = 1 MHz, TCC choice = 4 points frequency, TCC initial value = 156;

$$\text{TCC timingtime} = (1) \times (4) \times (256 - 156) = 400\ \text{us}$$

## 3.9 PWM pulse width modulation

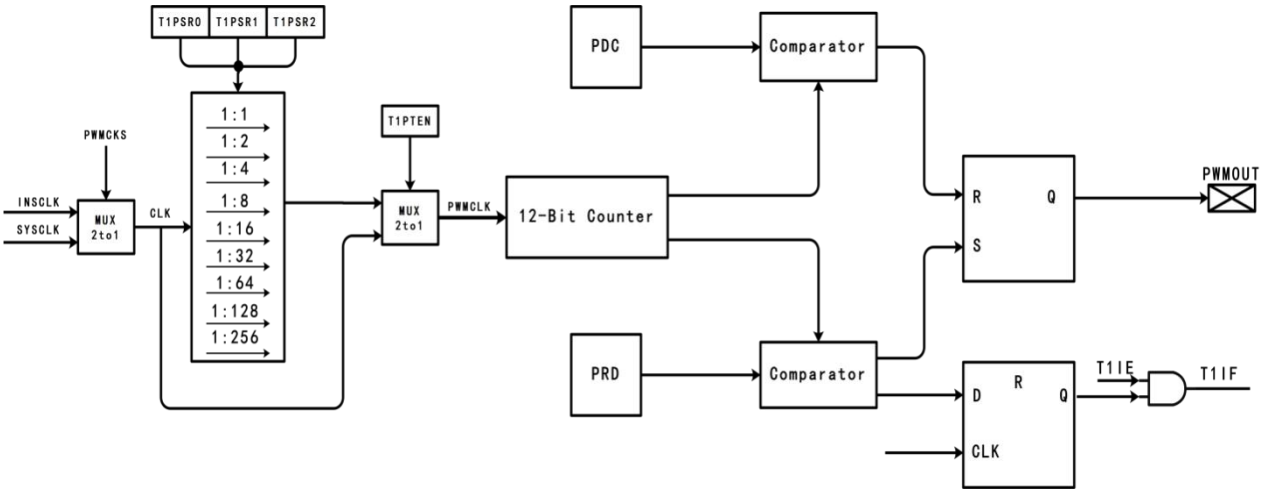
JZ8P1510 provide 1 road 12 bit PWM counter, which is used to produce PWM signal, PWM output waveform is determined by cycle and duty cycle, transfer rate of cycle count.

PWM with cycle overflow interrupt, when enabled can jump to interrupt.

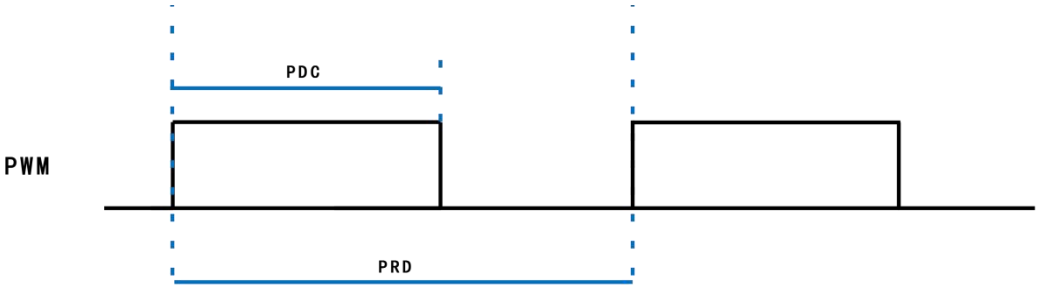
In IDLE (IDLE mode), in PWM mode control CPU registers PWMCKS = 1 and can make PWMWE, can wake up system.

### 3.9.1 PWM internal structure and sequence

The following chart is for reference only understand internal, does not represent the actual circuit.




PWM working structure circuit




PWM timing instructions

### 3.9.2 PWM cycle and duty cycle

PWM provides a 8 bit counter as a clock divider, provide a 12 bit counter (T1 counter) as a PWM module the baud rate of the clock generator. By enabling the T1EN PWM control register, can make the T1 counter function, through T1PTEN and T1PSR < > 2-0 control bits, T1 counter preassigned frequency can be set.

PWM cycle by writing register values to the PWM cycle (PRDCH, PRDL), when the T1 the value of the counter and PRD values are equal, increasing the next cycle the following events:

- T1 counter reset;
- Corresponding PWM output pin high level;
- Generate PWM cycle overflow interrupt (if enabled);
- PWM duty cycle by PRDCH, PDCL latch to PDC/T1 is register;

**PWM cycle calculation formula:**

$$\text{PWM cycle} = (\text{PRD}) \times (\frac{1}{Fosc}) \times (\text{T1Frequency division})$$

Example:

PRD = 100, Fosc = 8 MHz, T1 frequency selection = 2 points frequency;

$$\text{PWM cycle} = (100) \times \left(\frac{1}{8}\right) \times (2) = 25 \text{ us}$$

PWM duty cycle by writing values to the PWM duty cycle register (PRDCH, PDCL), when the T1 overflow reset the value of the counter, PRDCH, PDCL value is latched to the PDC/T1 compare register. When PDC/T1 more register values and T1 counter values are equal, PWM output pins for low level. PRDCH, PDCL values can be written at any time, but only in PDC/T1 more register values cycle overflow writes:

**PWM duty cycle calculation formula:**

$$\text{PWM Duty cycle} = (\text{PDC}) \times (\frac{1}{Fosc}) \times (\text{T1Frequency division})$$

Example:

PDC=50, Fosc=8 MHz, T1 divider selection =2 dividers;

$$\text{PWM Duty cycle} = (50) \times \left(\frac{1}{8}\right) \times (2) = 12.5 \text{ us}$$

### 3.9.3 PWM pulse width modulation setting instructions

- Set CPUCON register, can choose PWM invert output, PWM clock source and the function of PWM wake;
- Set PWMCON register, select the corresponding PWM mode, PWM frequency division ratio;
- Set the values of RPAG-R9, RA and RB registers to determine the cycle and duty cycle of the PWM channel.
- Enable PWM cycle overflow interrupt and execute "EI" or "DI" instruction (if necessary);

## 3.10 Wake up instructions

- **TCC wakes IDLE normally**
  - CPUCON register IDLE = 1, TCCWE = 1, TCCCKS = 1 (no need to interrupt when don't need to set up TCIE = 1);
  - To execute commands SLEEP;
- **TCC sensei IDLE low speed mode**
  - CPUCON register IDLE = 1, TCCWE = 1, TCCCKS = 1, CLKMD = 1, STPHX = 1 (no need to interrupt when don't need to set up TCIE = 1);
  - To execute commands SLEEP;
- **T1 under normal wake IDLE**
  - PWMCON register T1EN = 1;
  - Set the cycle register PRD;
  - CPUCON register IDLE = 1, PWMWE = 1, PWMCKS = 1 (no need to interrupt when don't need to set up T1IE = 1);
  - Execute the SLEEP command;
- **Wake IDLE in T1 low speed mode**
  - PWMCON register T1EN=1;
  - Set cycle register PRD;
  - CPUCON register IDLE = 1, PWMWE = 1, PWMCKS = 1, CLKMD = 1, STPHX = 1 (no need to interrupt when don't need to set up T1IE = 1);
  - Execute the SLEEP command;

# 4. OPTION configuration

| CODE OPTION | Options | Functional description |
|---|---|---|
| Clocks frequency division | 2 Clocks | The instruction cycle 2 Clocks |
| | 4 Clocks | Choose 4 Clocks for the instruction cycle |
| | 8 Clocks | Choose 8 Clocks for the instruction cycle |
| | 16 Clocks | Choose 16 Clocks for the instruction cycle |
| | 32 Clocks | Instruction cycle choose 32 Clocks |
| The IRC frequency | 8M | Choose 8M for IRC frequency |
| | 1M | Choose 1M for IRC frequency |
| | 910KHz | The IRC frequency selective 910 KHZ |
| The system clock frequency doubling option | prohibition | No doubling of the system clock is selected |
| | Can make | System clock select frequency multiplication (the system clock is a basic for the operation of chip internal clock, CPUCON registers the system clock can be configured for low speed or high speed mode, pay attention to and the oscillator frequency doubling distinguish) |
| Low voltage reset | As = 1.8 V | Choose 1.8V as the low voltage reset point |
| | As = 2.4 V | Low voltage reset point 2.4 V |
| | As = 2.7 V | Low voltage reset point 2.7 V |
| | As = 3.3 V | Low voltage reset point 3.3 V |
| | As = 3.6 V | Low voltage reset point 3.6 V |
| | As = 3.9 V | The low voltage reset point is chosen to be 3.9V |
| OTP paging choice | Use 1K | The OTP ROM can use the full 1K space |
| | Use the first 0.5K | OTP ROM can use 0.5 K space |
| | 0.5 K after use | OTP ROM can be used after 0.5 K space |
| Code encryption | Enable | Burn mode data encryption |
| | ban | Burn mode data is not encrypted |
| Reset on port | Enable | Enable P63 port pull-up function |
| | prohibition | Ban about port function |
| About the port | GPIO | P63 serves as a general purpose I/O port |
| | GPI | P63 as input port |
| | RST | About as an external reset port |

| | | |
|---|---|---|
| Reset time | PWRT = 4.5 ms | Reset setup time =4.5ms |
| | PWRT=18ms | Reset setup time =18ms |
| | PWRT=72ms | Reset time = 72 ms |
| | PWRT=288ms | Reset setup time =288ms |
| | PWRT=140us | Reset setup time =140us |
| Scope of look-up table | 1/4K | Can look up tables 1/4K in front of program space |
| | 1K | Can look up table within 1 k application space |
| P6 port awaken | Independent control | Zero x0d P6 port independently controlled wake up (register) |
| | The independent control | P6 port not awaken the independent control |
| P6 pull-down control | shielding | Shield P65-P64 pull-down |
| | Enable | Enable P65-P64 pull-down, P63 pull-up software control |
| POWER | HIGH | Select HIGH when the VDD voltage works around 5V (pay attention to the change of voltage frequency characteristics) |
| | LOW | Work at about 3 v LOW VDD voltage (note the change of pressure frequency characteristics) |

# 5. Instruction set

| Instructions | Command action | Flag impact |
|---|---|---|
| ADD A,R | A+R→A | Z,C,DC |
| ADD R,A | A+R→R | Z,C,DC |
| AND A,R | A&R→A | Z |
| AND R,A | A&R→R | Z |
| CLRA | 0→A | Z |
| CLR R | 0→R | Z |
| INVA R | /R→A | Z |
| INV R | /R→R | Z |
| DA | A register adjusts to BCD value | C |
| DECA R | R-1→A | Z |
| DEC R | R-1→R | Z |
| DJA R | R-1→A, skip if zero | - |
| DJ R | R-1→R, skip if zero | - |
| INCA R | R+1→A | Z |
| INC R | R+1→R | Z |
| IJA R | R+1→A, skip if zero | - |
| IJ R | R+1→R, skip if zero | - |
| MOV R,A | A→R | - |
| MOV A,R | R→A | Z |
| MOV R,R | R→R | Z |
| OR A,R | A∨VR→A | Z |
| OR R,A | A∨VR→R | Z |
| SUB A,R | R-A→A | Z,C,DC |
| SUB R,A | R-A→R | Z,C,DC |
| XOR A,R | A⊕R→A | Z |
| XOR R,A | A⊕R→R | Z |
| IR R | IOCR→A | - |
| IW R | A→IOCR | - |
| CTR | CONT→A | - |
| CTW | A→CONT | - |
| BTC R,b | 0→R(b) | - |
| BTS R,b | 1→R(b) | - |

| | | |
|---|---|---|
| JBTC R,b | if R(b)=0, skip | - |
| JBTS R,b | if R(b)=1, skip | - |
| LCR R | R(n)→R(n+1),R(7)→C, C→R(0) | C |
| LCA R | R(n)→A(n+1),R(7)→C, C→A(0) | C |
| RCR R | R(n)→R(n-1),R(0)→C, C→R(7) | C |
| RCA R | R(n)→A(n-1),R(0)→C, C→A(7) | C |
| SWAP R | R(0-3) ↔ R(4-7) | - |
| SWAPA R | R(0-3)→A(4-7),R(4-7)→A(0-3) | - |
| ADD A,k | A+k→A | Z,C,DC |
| AND A,k | A&k→A | Z |
| MOV A,k | k→A | - |
| OR A,k | A∨k→A | Z |
| SUB A,k | k-A→A | Z,C,DC |
| XOR A,k | A⊕k→A | Z |
| CALL k | PC+1→[SP], (Page,k)→PC | - |
| DI | Disallow interruption | - |
| EI | Enable Interrupt | - |
| JMP k | K (Page,k)→PC | - |
| NOP | Empty instruction | - |
| RET | [top of stack]→PC | - |
| RETI | [top of stack]→PC, enable interrupt | - |
| RETL k | k→A,[top of stack]→PC | - |
| SLEEP | Oscillator stops oscillating (sleep mode) | T,P |
| CWDT | 0→WDT | T,P |

# 6. Electrical characteristics

## 6.1 Limit parameters

Working temperature..... -40℃ ~ 85℃
Storage temperature..... -65℃ to 150℃
The input voltage..... Vss-0.3V ~ Vdd+0.5V
The output voltage..... Vss-0.3V ~ Vdd+0.5V
Working voltage..... 1.8 V to 5.5 V

## 6.2 DC electrical characteristics

**($V_{DD}$ =5V, operating temperature =25 ° C, unless otherwise stated)**

| Symbol | Parameter description | Conditions | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| IRC1 | IRC1 (after correction) | OPTION select 8MHz | - | 8 | - | MHz |
| IRC2 | IRC2 (after correction) | OPTION select 1MHz | - | 1 | - | MHz |
| IRC3 | IRC3 (after correction) | OPTION select 910KHz | - | 910 | - | KHz |
| IOH1 | IO output high level drive (except P63) | Ioh = 4.4 V | 10 | 11 | 12 | mA |
| IOH2 | IO output High level driver (P63) | Ioh = 4.4 V | 5 | 5.5 | 6 | mA |
| IOL1 | IO output low level driver (except P63) | Iol = 0.6 V | 13 | 14 | 15 | mA |
| IOL2 | IO output low level driver (P63) | Iol = 0.6 V | 11 | 12 | 13 | mA |
| IPH1 | Pull-up current (except P63) | Pull-up enable, input ground | 90 | 95 | 100 | μA |
| IPH2 | Pull-up current (P63) | Pull-up enable, input ground | 110 | 120 | 130 | μA |
| IPD | Pull-down current (except P63) | Pull-down enabled, input to VDD | 45 | 50 | 55 | μA |
| Isb1 | Power off current 1 | The input is connected to VDD, and the output is suspended | - | - | 1 | μA |
| Iop1 | Operating current 1 (VDD=5V) | IRC=8MHz 2clock | - | 1.2 | 1.5 | mA |
| Iop2 | Operating current 2 (VDD=5V) | IRC=910KHz 2clock | - | 0.2 | 0.3 | mA |
| Iop3 | Operating current 3 (low speed mode) (VDD=5V) | System clock Select low speed clock | - | 5 | 8 | μA |
| LVR | Low voltage reset voltage | Select the LVR reset point | Vlvr - 0.2 | Vlvr | Vlvr +0.2 | V |
| ILRC | Low speed oscillator frequency | - | 10K | - | 25K | Hz |

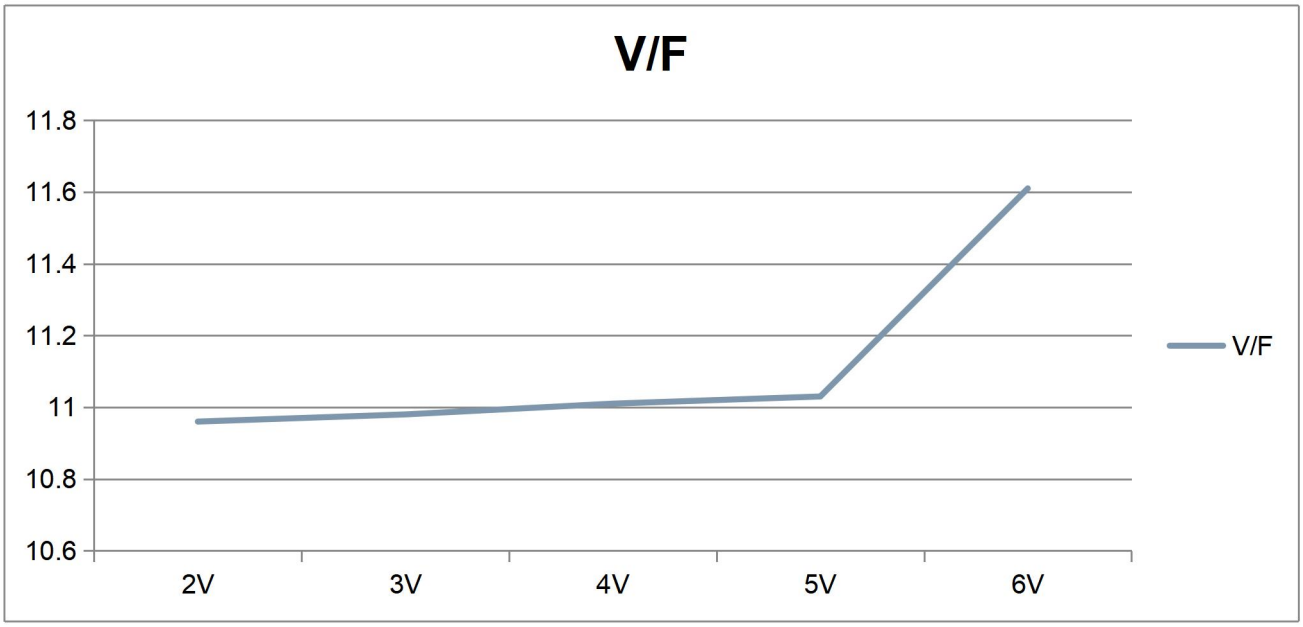
Note: The above parameters are for reference only, please refer to the measured data of the target prototype.

## 6.3 Characteristic curve diagram

The curves listed in this chapter are for design reference only, and part of the data given may exceed the specified working range of the chip. In order to ensure the normal operation of the chip, please strictly refer to the electrical characteristics.

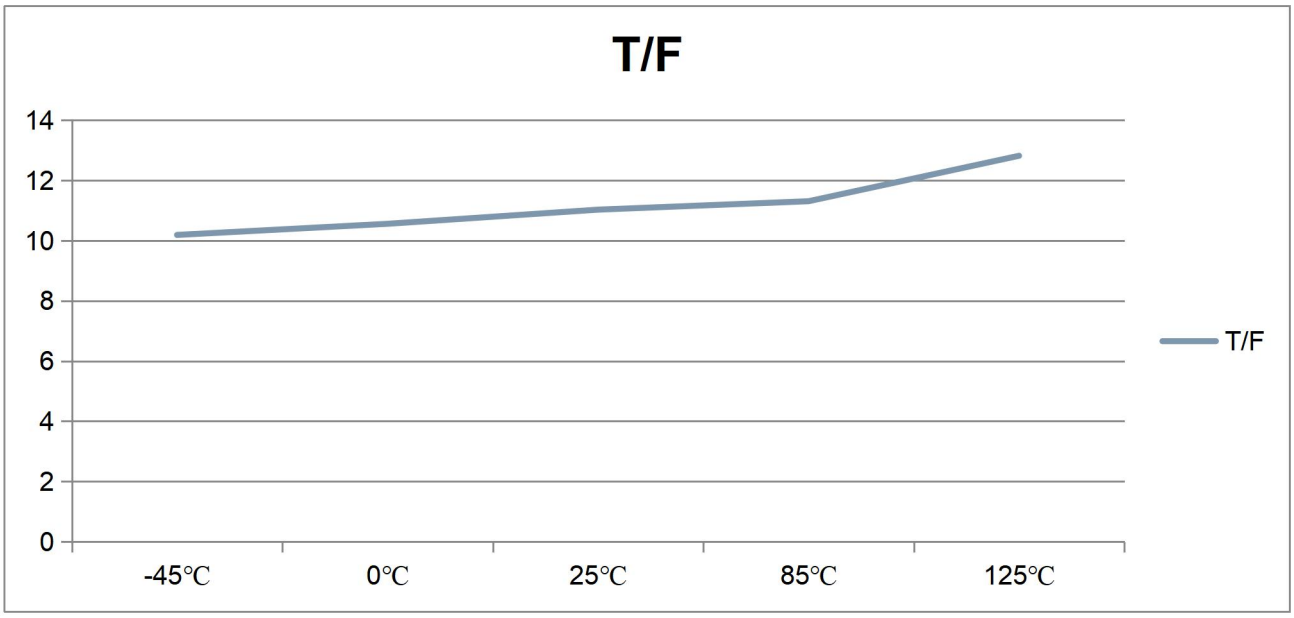
### 6.3.1 Internal low-speed RC Oscillator - voltage-frequency characteristic curve

Operating temperature at 25℃ : (in KHz)



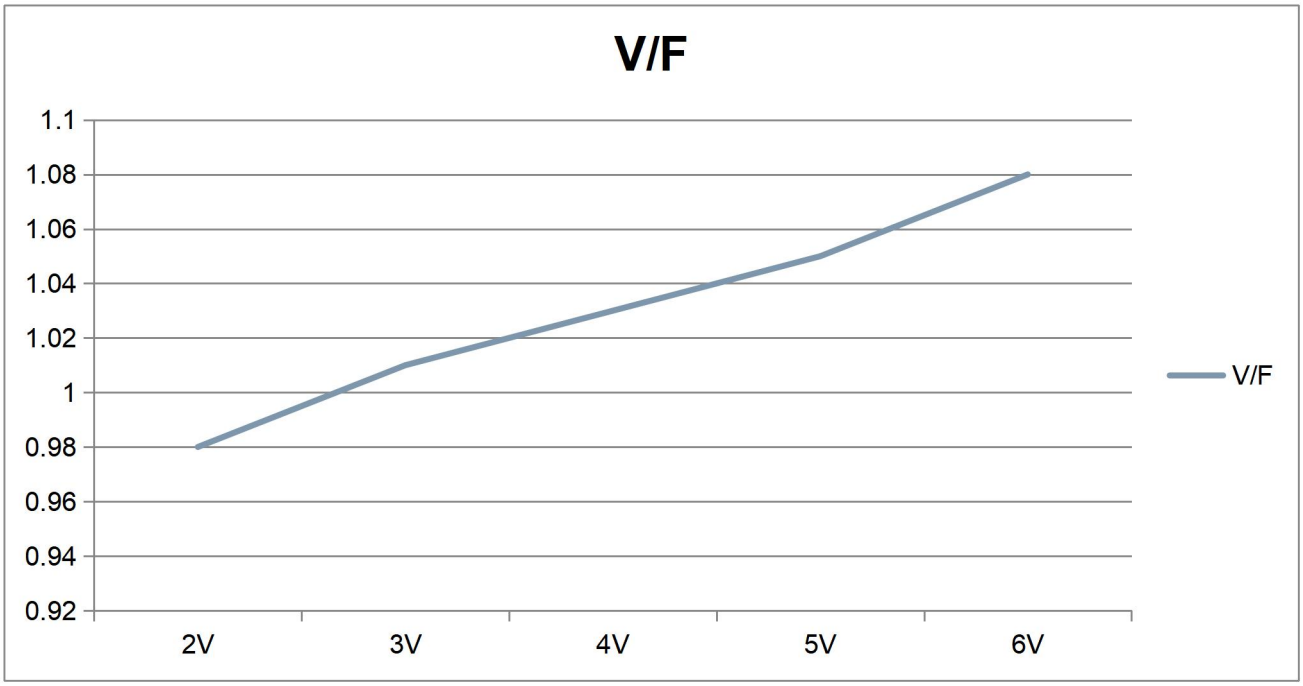

### 6.3.2 Internal low-speed RC oscillator - Temperature-frequency characteristic curve

Operating voltage at 5V: (in KHz)



### 6.3.3 Internal 1MHz RC oscillator - voltage-frequency characteristic curve

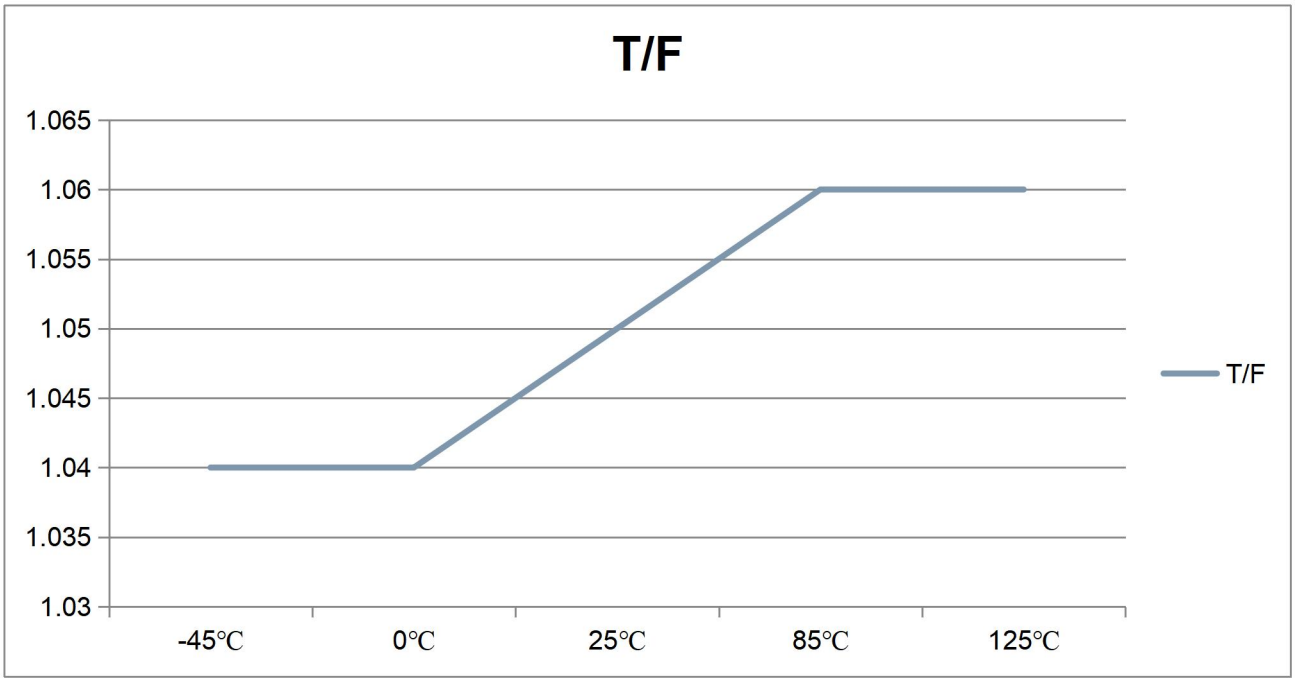
Operating temperature at 25℃ : (in MHz)




]

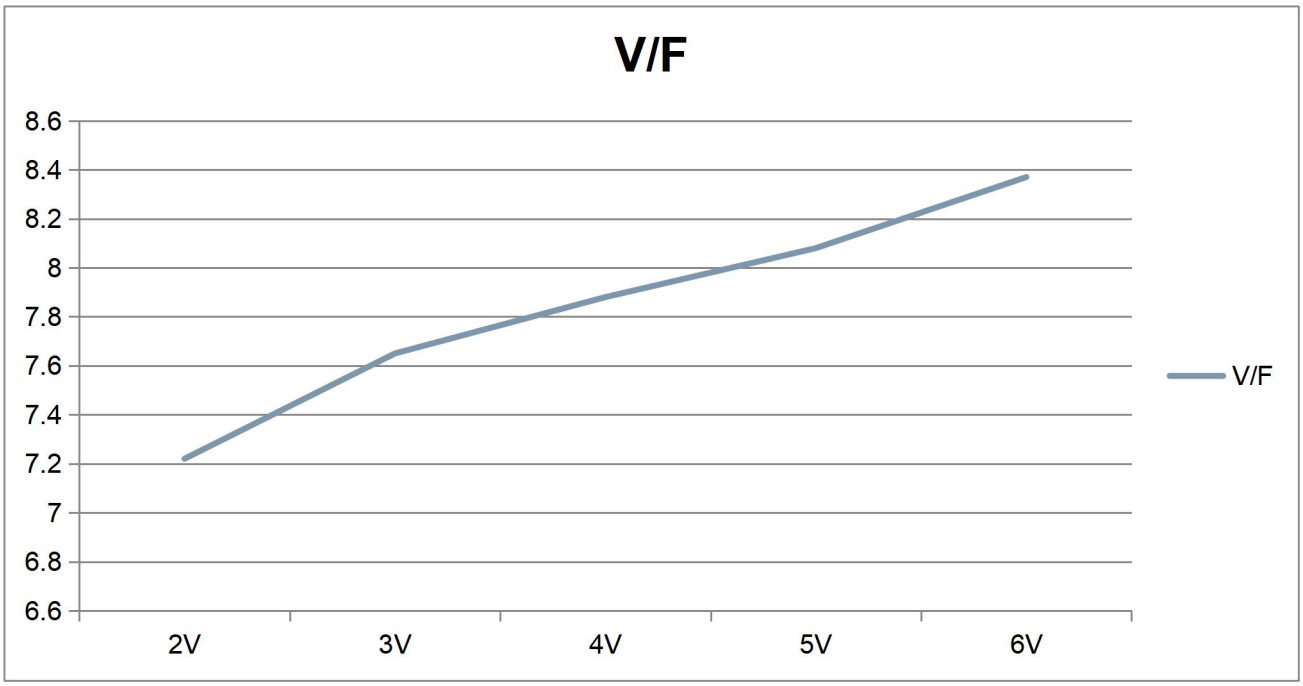
### 6.3.4 Internal 1MHz RC oscillator - Temperature-frequency characteristic curve

Operating voltage at 5V: (unit MHz)



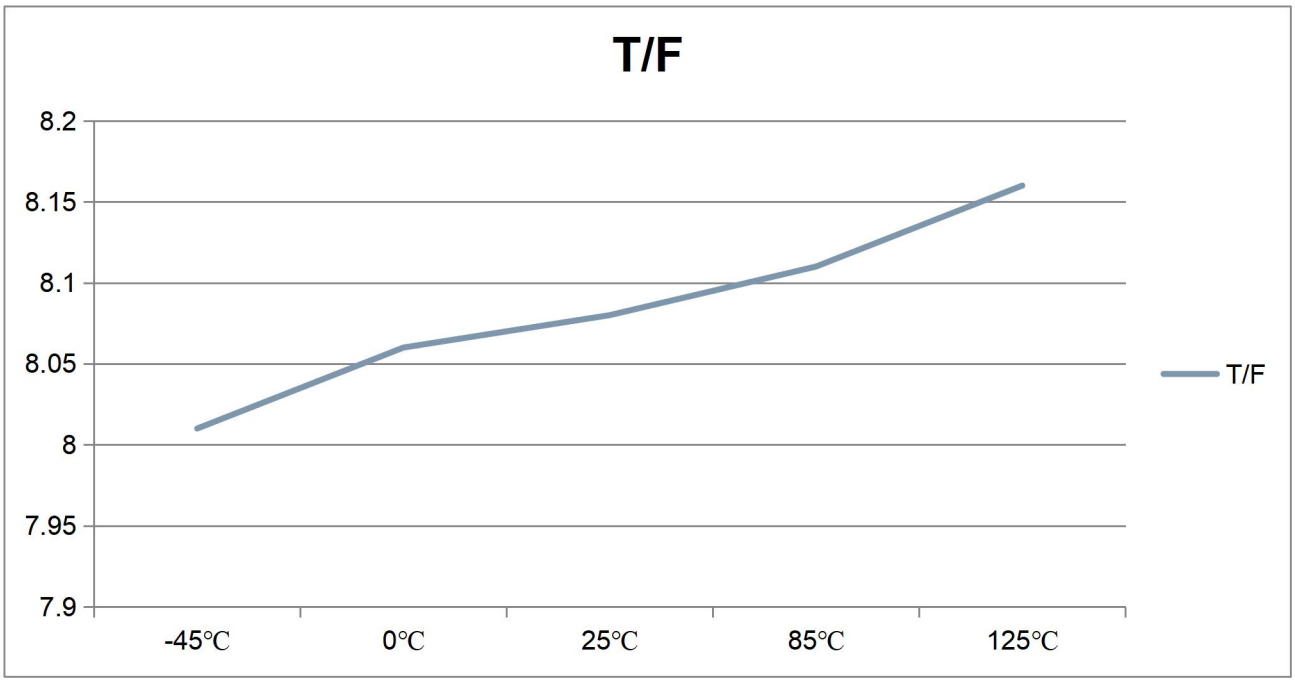
### 6.3.5 Internal 8MHz RC oscillator - voltage-frequency characteristic curve

Operating temperature at 25 ° C: (in MHz)




### 6.3.6 Internal 8MHz RC oscillator - Temperature-frequency characteristic curve

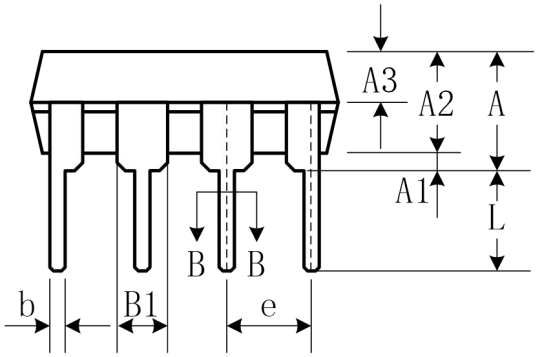
Operating voltage at 5V: (in MHz)

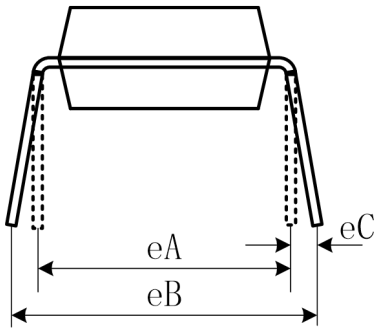


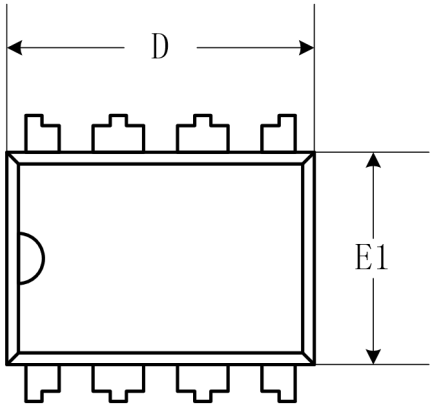
# 7 Package size

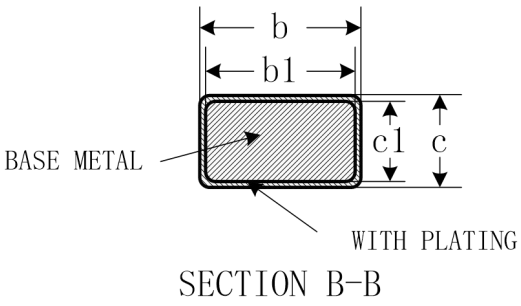
## 7.1 8PIN package size

**DIP8**



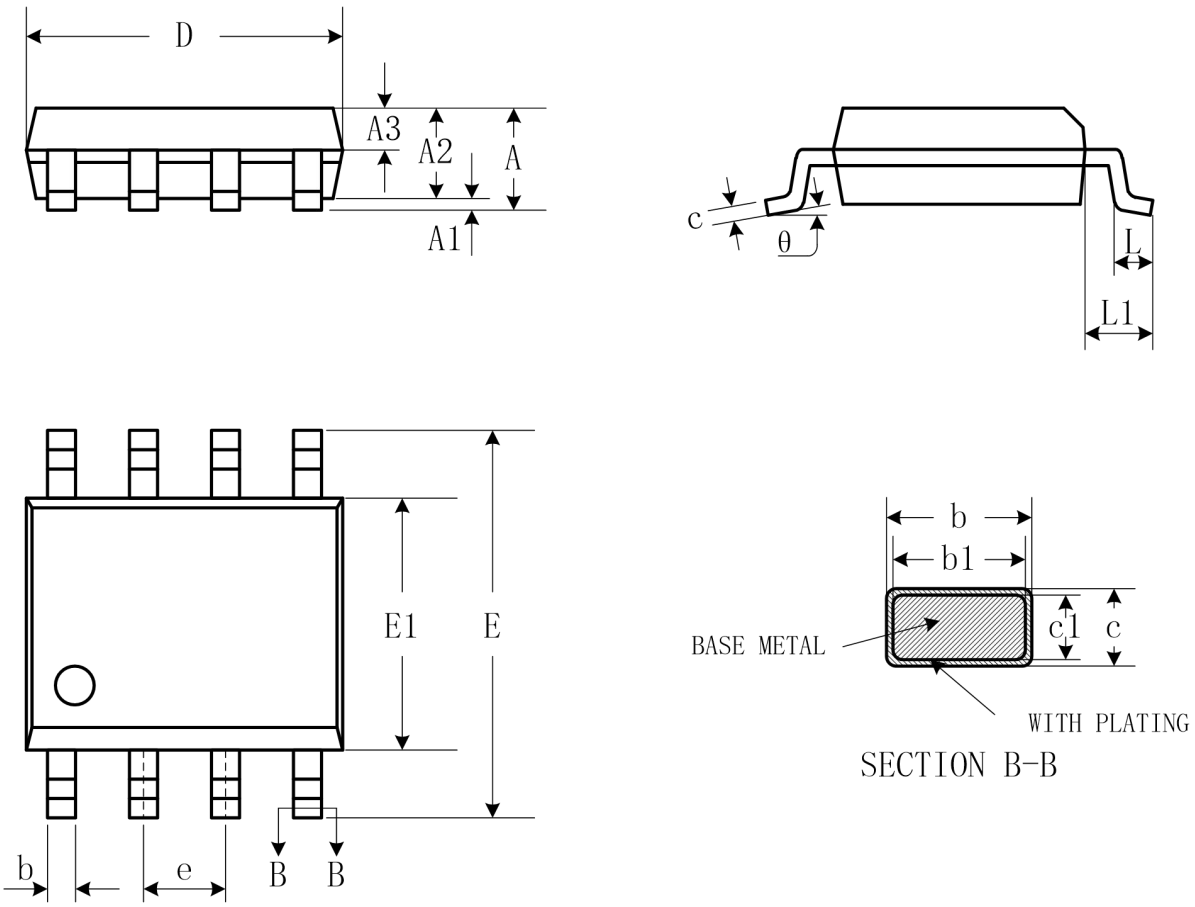




| SYMBOL | MILLIMETER | | |
|---|---|---|---|
| | MIN | NOM | MAX |
| A | 3.60 | 3.80 | 4.00 |
| A1 | 0.51 | - | - |
| A2 | 3.10 | 3.30 | 3.50 |
| A3 | 1.50 | 1.60 | 1.70 |
| b | 0.44 | - | 0.53 |
| b1 | 0.43 | 0.46 | 0.48 |
| B1 | | 1.52 BSC | |
| c | 0.25 | - | 0.31 |
| c1 | 0.24 | 0.25 | 0.26 |
| D | 9.05 | 9.25 | 9.45 |
| E1 | 6.15 | 6.35 | 6.55 |
| e | | 2.54 BSC | |
| eA | | 7.62 BSC | |
| eB | 7.62 | - | 9.50 |
| eC | 0 | - | 0.94 |
| L | 3.00 | - | - |

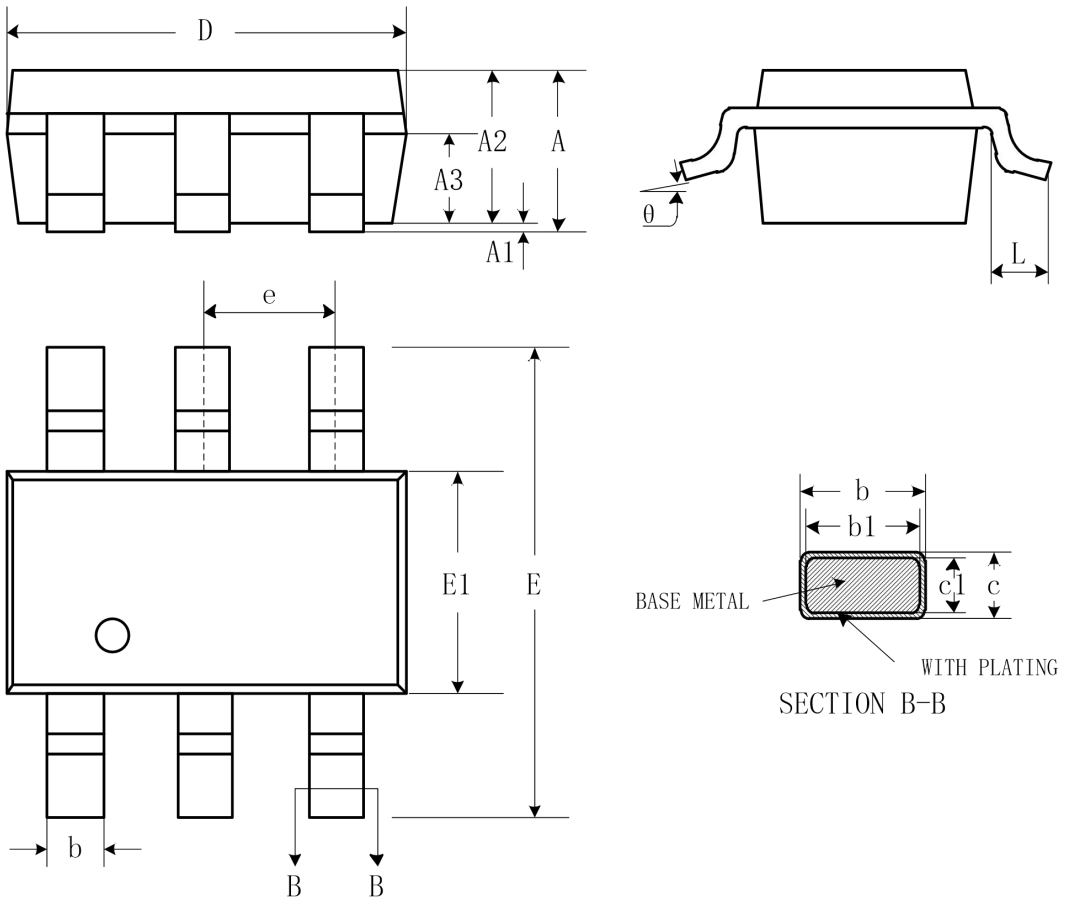
**SOP8**




| SYMBOL | MILLIMETER | | |
|---|---|---|---|
| | MIN | NOM | MAX |
| A | - | - | 1.77 |
| A1 | 0.08 | 0.18 | 0.28 |
| A2 | 1.20 | 1.40 | 1.60 |
| A3 | 0.55 | 0.65 | 0.75 |
| b | 0.39 | - | 0.48 |
| b1 | 0.38 | 0.41 | 0.43 |
| c | 0.21 | - | 0.26 |
| c1 | 0.19 | 0.20 | 0.21 |
| D | 4.70 | 4.90 | 5.10 |
| E | 5.80 | 6.00 | 6.20 |
| E1 | 3.70 | 3.90 | 4.10 |
| e | 1.27 BSC | | |
| L | 0.50 | 0.65 | 0.80 |
| L1 | 1.05 BSC | | |
| θ | 0 | - | 8 ° |

## 7.2 6PIN package size

**SOT23-6**




**Package Specifications:**

| Symbol | mm | | | Inch | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOR | MAX |
| A | — | — | 1.35 | — | — | 0.053 |
| A1 | 0.04 | — | 0.15 | 0.002 | — | 0.006 |
| A2 | 1.00 | 1.10 | 1.20 | 0.039 | 0.043 | 0.047 |
| A3 | 0.55 | 0.65 | 0.75 | 0.022 | 0.026 | 0.030 |
| b | 0.30 | — | 0.50 | 0.013 | — | 0.017 |
| b1 | 0.30 | 0.40 | 0.45 | 0.013 | 0.016 | 0.018 |
| c | 0.08 | — | 0.22 | 0.006 | — | 0.008 |
| c1 | 0.08 | 0.13 | 0.20 | 0.003 | 0.005 | 0.08 |
| D | 2.72 | 2.92 | 3.12 | 0.107 | 0.115 | 0.123 |
| E | 2.60 | 2.80 | 3.00 | 0.102 | 0.110 | 0.118 |
| E1 | 1.40 | 1.60 | 1.80 | 0.055 | 0.063 | 0.071 |
| e | 0.95 BSC | | | 0.037 BSC | | |
| L | 0.30 | — | 0.60 | 0.012 | — | 0.024 |
| θ | 0 | — | 8° | 0 | — | 8° |