JZ8P2612

8 bit OTP Microcontroller

USER’S MANUAL

Version V1.1

Revision History

Version	Description	Note
V1.0	First issue	
V1.1	Modifying the Content Format	

Statement：

- This information will be updated with the improvement of the product without prior notice.

  Please consult our sales staff before using this information to ensure that the content of this information is the latest version.

- Please use this product within the limit specified in this document. We will not be responsible for any loss caused by improper use.
- Although the company has always been committed to improve the quality and reliability of products, but the semiconductor products themselves have a certain probability of failure or wrong work, in order to prevent such accidents caused by personal injury or property loss, please pay full attention to the safety design during the use.
- When exporting this product or this data overseas, it shall comply with applicable laws and regulations on import and export control.
- Without the permission of the Company, it is strictly prohibited to copy or reprint this information in part or in whole in any form.
- The test data are for reference only. The actual data are subject to the target prototype test.

CONTENTS

1 PRODUCT OVERVIEW 7
1.1 FEATURES 7
1.2 PIN ASSIGNMENT 8
1.3 PIN DESCRIPTION 9
1.4 BLOCK DIAGRAM 10
2 MEMORY STRUCTURE 10
2.1 PROGRAM MEMORY 10
2.2 DATA MEMORY 11
2.2.1 DATA REGISTER AREA (RAM) 11
2.2.2 SYSTEM REGISTER TABLE 11
2.2.3 SYSTEM REGISTER DESCRIPTION 12
2.2.4 BIT DEFINITION OF SYSTEM REGISTER 13
3 FUNCTIONAL DESCRIPTION 14
3.1 SYSTEM REGISTER 14
3.1.1 082H-R (R REGISTER) 14
3.1.2 083H-Z (Z REGISTER) 14
3.1.3 084H-Y (Y REGISTER) 14
3.1.4 086H-PFLAG (PROGRAM FLAG) 14
3.1.5 087H-IRCCAL (IRC FREQUENCY ADJUSTMENT REGISTER) 15
3.1.6 0AEH-P4CON (P4 CONFIGURATION REGISTER) 15
3.1.7 0AFH-VREF (ADC REFERENCE VOLTAGE REGISTER) 15
3.1.8 0B1H-ADM (ADC MODE REGISTER) 16
3.1.9 0B2H-ADB (ADC DATA BUFFER REGISTERS) 17
3.1.10 0B3H-ADR (ADC PRECISION SELECTION REGISTER) 17
3.1.11 0B4H-ADT (ADC ZERO DRIFT REGISTER) 17
3.1.12 0B8H-P0M (P0 PORT MODE) 17
3.1.13 0BFH-PEDGE (EXTERNAL INTERRUPT OPERATION) 17
3.1.14 0C4H-P4M (P4 PORT MODE) 17
3.1.15 0C5H-P5M (P5 PORT MODE) 18
3.1.16 0C8H-INTRQ (INTRQ INTERRUPT REQUEST REGISTER) 18
3.1.17 0C9H-INTEN (INTEN INTERRUPT ENABLE REGISTER) 18
3.1.18 0CAH-OSCM (OSCM REGISTER) 18
3.1.19 0CCH-WDTR (WATCHDOG CLEAR REGISTER) 19
3.1.20 0CDH-TCOR (TCOR AUTO-LOAD REGISTER) 19
3.1.21 0CEH-PCL (PROGRAM COUNTER LOW-BYTE) 19
3.1.22 0CFH-PCH (PROGRAM COUNTER HIGH-BYTE) 19
3.1.23 0D0H-P0 (P0 PORT DATA REGISTER) 20
3.1.24 0D4H-P4 (P4 PORT DATA REGISTER) 20
3.1.25 0D5H-P5 (P5 PORT DATA REGISTER) 20
3.1.26 0D8H-T0M (TC0X8, TC0GN FLAGS) 20
3.1.27 0DAH-TCOM (TCOM MODE REGISTER) 20
3.1.28 0DBH-TC0C (TC0C COUNTING REGISTER) 21

3.1.29 ODCH-TC1M（TC1M MODE REGISTER）.....22
3.1.30 ODDH-TC1C（TC1C COUNTING REGISTER）.....22
3.1.31 ODEH-TC1R（TC1R AUTO-LOAD REGISTER）.....23
3.1.32 ODFH-STKP（STACK REGISTERS）.....23
3.1.33 OE0H-P0UR（P0 PULL UP CONTROL REGISTER）.....24
3.1.34 OE4H-P4UR（P4 PULL UP CONTROL REGISTER）.....24
3.1.35 OE5H-P5UR（P5 PULL UP CONTROL REGISTER）.....24
3.1.36 OE7H-@YZ（@YZ REGISTER）.....24
3.1.37 OF8H～OFFH-STK0～STK3（STACK REGISTERS）.....24
3.2 GPIO PORT.....25
3.2.1 GPIO REGISTER SPECIFICATION.....25
OB8H（P0M PORT MODE）.....25
0C4H（P4M PORT MODE）.....25
0C5H（P5M PORT MODE）.....25
0D0H（P0 PORT DATA REGISTER）.....25
0D4H（P4 PORT DATA REGISTER）.....25
0D5H（P5 PORT DATA REGISTER）.....26
0E0H（P0UR PULL UP CONTROL REGISTER）.....26
0E4H（P4UR PULL UP CONTROL REGISTER）.....26
0E5H（P5UR PULL UP CONTROL REGISTER）.....26
3.3 TIMERS.....27
3.3.1 WATCHDOG TIMER.....27
0CCH（WDTR WATCHDOG CLEAR REGISTER）.....27
3.3.2 TIMER/COUNTER (TC0).....27
3.3.2.1 OVERVIEW.....27
3.3.2.2 TC0 TIMER OPERATION.....28
3.3.2.3 TC0M MODE REGISTER.....30
3.3.2.4 TC0X8/TC0GN FLAGS.....30
3.3.2.5 TC0C COUNTING REGISTER.....31
3.3.2.6 TC0R AUTO-LOAD REGISTER.....31
3.3.2.7 TC0 EVENT COUNTER.....32
3.3.2.8 TC0 CLOCK FREQUENCY OUTPUT.....32
3.3.2.9 PULSE WIDTH MODULATION(PWM).....33
3.3.3 TIMER/COUNTER 1 (TC1).....34
3.3.3.1 OVERVIEW.....34
3.3.3.2 TC1 TIMER OPERATION.....34
3.3.3.3 TC1M MODE REGISTER.....36
3.3.3.4 TC1X8/TC1GN FLAG.....36
3.3.3.5 TC1C COUNTING REGISTER.....37
3.3.3.6 TC1R AUTO-LOAD REGISTER.....37
3.3.3.7 TC1 EVENT COUNTER FUNCTION.....38
3.3.3.8 TC1 CLOCK FREQUENCY OUTPUT.....38
3.3.3.9 PULSE WIDTH MODULATION (PWM).....39
3.4 ANALOG TO DIGITAL CONVERTER.....41
3.4.1 OVERVIEW.....41

3.4.2 ADC MODE REGISTER 41
0B1H（ADM CONTROL REGISTER） 41
0B2H（ADC DATA BUFFER REGISTERS） 42
0B3H（ADC PRECISION SELECTION REGISTER） 42
3.4.3 ADC PIN CONFIGURATION 42
0AEH（P4CON PORT CONFIGURATION CONTROL REGISTER） 43
3.4.4 ADC REFERENCE VOLTAGE REGISTER 43
0AFH（ADC REFERENCE VOLTAGE REGISTER） 43
3.4.5 ADC APPLICATION CIRCUIT 44
3.4.6 ADC OPERATION EXAMLPE 44
3.5 LOW VOLTAGE DETECTOR 46
3.5.1 LVD VOLTAGE DETECTION REGISTER 46
086H（PFLAG PROGRAM FLAG） 46
3.6 INTERRUPT 47
3.6.1 INTERRUPT ENABLE REGISTER 47
0C9H（INTEN INTERRUPT ENABLE REGISTER） 47
3.6.2 INTERRUPT REQUEST REGISTER 47
0C8H（INTRQ INTERRUPT REQUEST REGISTER） 48
3.6.3 GIE GLOBAL INTERRUPT OPERATION 48
0DFH（STKP CONTROL REGISTER） 48
3.6.4 INT0（P0.0）EXTERNAL INTERRUPT OPERATION 48
0BFH（PEDGE CONTROL REGISTER） 48
3.6.5 INT1（P0.1）INTERRUPT OPERATION 49
3.6.6 TC0 INTERRUPT 49
3.6.7 TC1 INTERRUPT 49
3.6.8 ADC INTERRUPT 50
3.6.9 MULTI-INTERRUPT OPERATION 50
3.7 RESET 52
3.7.1 OVERVIEW 52
0B6H（PFLAG CONTROL REGISTER） 52
3.7.2 POWER ON RESET 53
3.7.3 WATCHDOG RESET 53
3.7.4 BROWN OUT RESET 54
3.7.5 THE SYSTEM OPERATING VOLTAGE 55
3.8 SYSTEM CLOCK 56
3.8.1 EXTERNAL CRYSTAL OSCILLATOR 56
3.8.2 INTERNAL RC OSCILLATOR MODE（IRC） 57
3.8.3 OSCM MODE REGISTER 57
3.9 SYSTEM OPERATION MODE 58
3.9.1 OVERVIEW 58
3.9.2 NORMAL MODE 58
3.9.3 SLOW MODE 59
3.9.4 SLEEP MODE 59
3.9.5 GREEN MODE 60
3.9.6 OPERATING MODE CONTROL MACRO 61

3.9.7 WAKEUP 61
4 CODE OPTION REGISTER 63
5 ELECTRICAL CHARACTERISTIC 64
5.1 ABSOLUTE MAXIMUM RATING 64
5.2 DC CHARACTERISTIC 64
6 PACKAGE INFORMATION 65
6.1 16PIN PACKAGE INFORMATION 65
6.2 14PIN PACKAGE INFORMATION 66
6.3 8PIN PACKAGE INFORMATION 67
6.4 6PIN PACKAGE INFORMATION 68

1 PRODUCT OVERVIEW

1.1 FEATURES

CPU Configuration

- 2K×16-Bit OTP ROM
- 128×8-Bit SRAM
- 4 -level stacks
- 2 -level programmable LVD 2.4V/3.6V
- 3 -level programmable LVR 2.0V/2.4V/3.6V
- Operating current less than 1mA （16MHz/5V）
- Operating current 5μA（32kHz/3V）
- Sleep mode current is less than 1μA

I/O Port Configuration

- 3 bidirectional I/O:P0,P4,P5
- 14 I/O pins
  (P04 only drain open output)
- Wake-up port:P00～P06
  （level trigger）
- 13 programmable pull-up I/O pins
  (P04 pull-up can be configured through OPTION)
- External interrupt:P0.0, P0.1

Operating Voltage

- Operating voltage range:
  2V～5.5V (0℃-70℃)
  2.3V～5.5V (-40℃-85℃)

System Clock

- Internal low speed oscillation circuit:
  16KHz（3V）
  32KHz (5V)
- Internal high speed oscillation circuit:
  4MHz,8MHz,16MHz,32MHz
- External XT Oscillator Circuit:
  16MHz
- Clock Cycle Frequency Division:
  8T/16T/32T/64T/128T

Peripheral configuration

- 5 channel ADC with 12 bit resolution
- 1 internal battery measurement
- Internal AD reference voltage （2V,3V,4V）
- 8Bit*2 timer /counter
- 8Bit*2 PWM/Buzzer output

Available Interrupts

- 3 internal interrupts TC0/TC1/ADC
- 2 external interrupt INT0/INT1

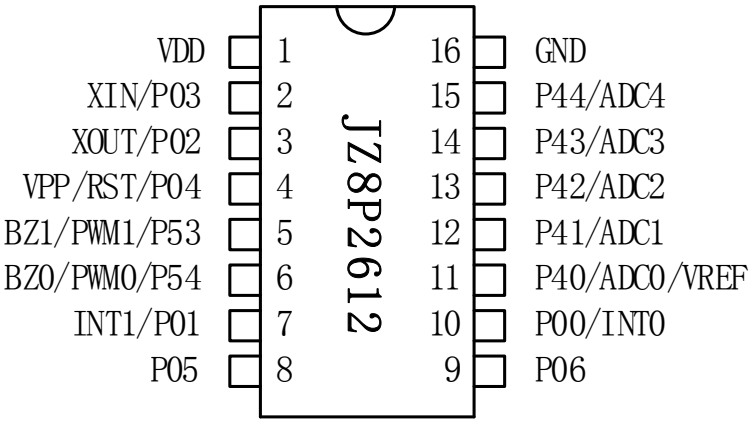
Operating Mode

- Normal mode: Both high and low clock active
- Slow mode: Low clock only
- Sleep mode: Both high and low clock stop
- Green mode: Periodical wakeup by TC0 timer

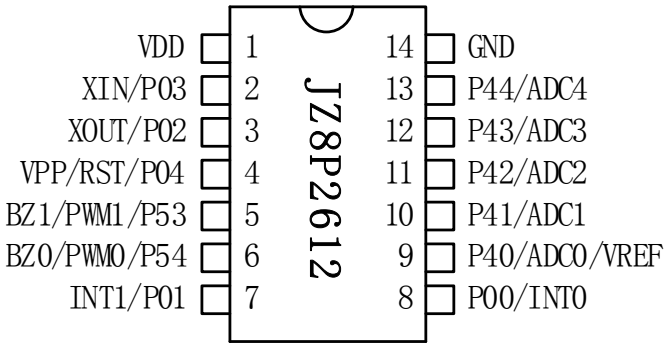
Package Type

- JZ8P2612-SOP16;
- JZ8P2612-SOP14;
- JZ8P2612-SOP8;
- JZ8P2612-DIP16;
- JZ8P2612-DIP14;
- JZ8P2612-DIP8;
- JZ8P2612-SOT23-6;

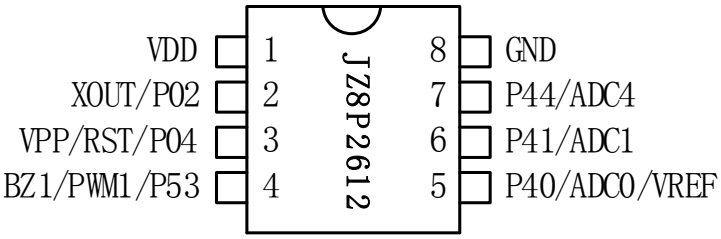
1.2 PIN ASSIGNMENT



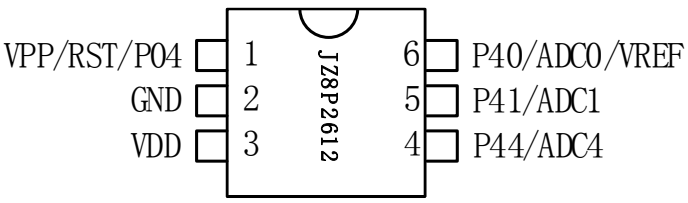
JZ8P2612-16PIN



JZ8P2612-14PIN



JZ8P2612-8PIN

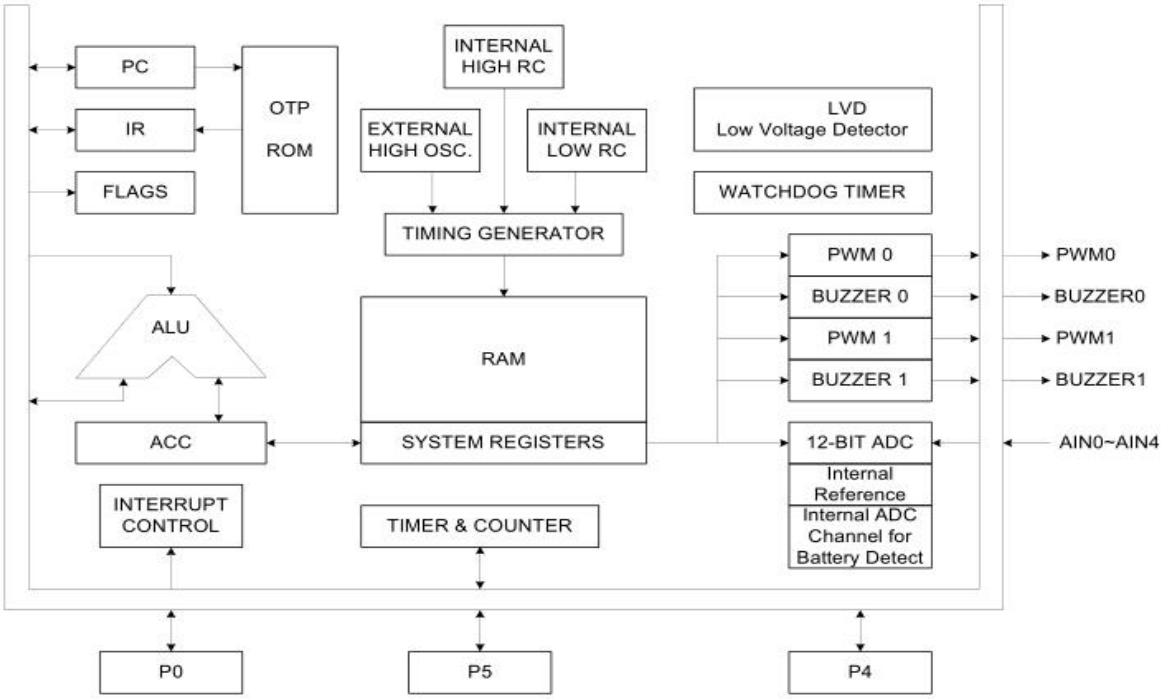


JZ8P2612-6PIN

1.3 PIN DESCRIPTION

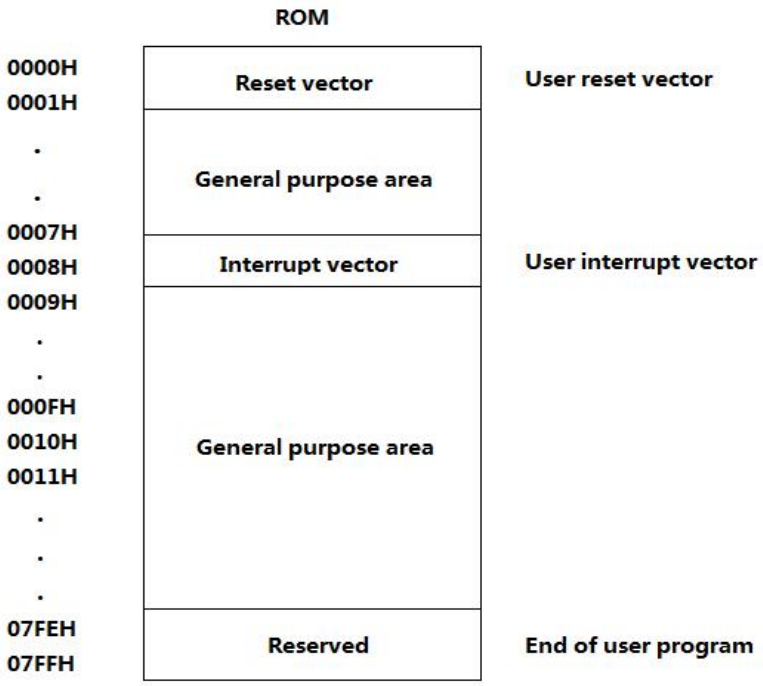
Name	Function	I/O	Description
P03	P03	I/O	GPIO, programmable pull-up、port change wake-up
	XIN	I	External oscillator input port
P02	P02	I/O	GPIO, programmable pull-up、port change wake-up
	XOUT	O	External oscillator output port
P04	P04	I/O	GPI, port change wake-up, drain open output, pull-up (OPTION configuration pull up)
	RST	I	External reset input port, low level trigger
	VPP	I	External programming high voltage input port
P53	P53	I/O	GPIO, programmable pull-up
	PWM1	O	PWM1 output
	BZ1	O	Buzzer1 output
P54	P54	I/O	GPIO, programmable pull-up
	PWM0	O	PWM0 output
	BZ0	O	Buzzer0 output
P01	P01	I/O	GPIO, programmable pull-up、port change wake-up
	INT1	I	External interrupt port
P05	P05	I/O	GPIO, programmable pull-up
P06	P06	I/O	GPIO, programmable pull-up
P00	P00	I/O	GPIO, programmable pull-up、port change wake-up
	INT0	I	External interrupt port
P40	P40	I/O	GPIO, programmable pull-up
	ADC0	I	ADC external input detection port
	VREF	I	ADC external reference voltage input port
P41	P41	I/O	GPIO, programmable pull-up
	ADC1	I	ADC input detection port
P42	P42	I/O	GPIO, programmable pull-up
	ADC2	I	ADC input detection port
P43	P43	I/O	GPIO, programmable pull-up
	ADC3	I	ADC input detection port
P44	P44	I/O	GPIO, programmable pull-up
	ADC4	I	ADC input detection port
	VDD	--	Vdd
	GND	--	Gnd

1.4 BLOCK DIAGRAM



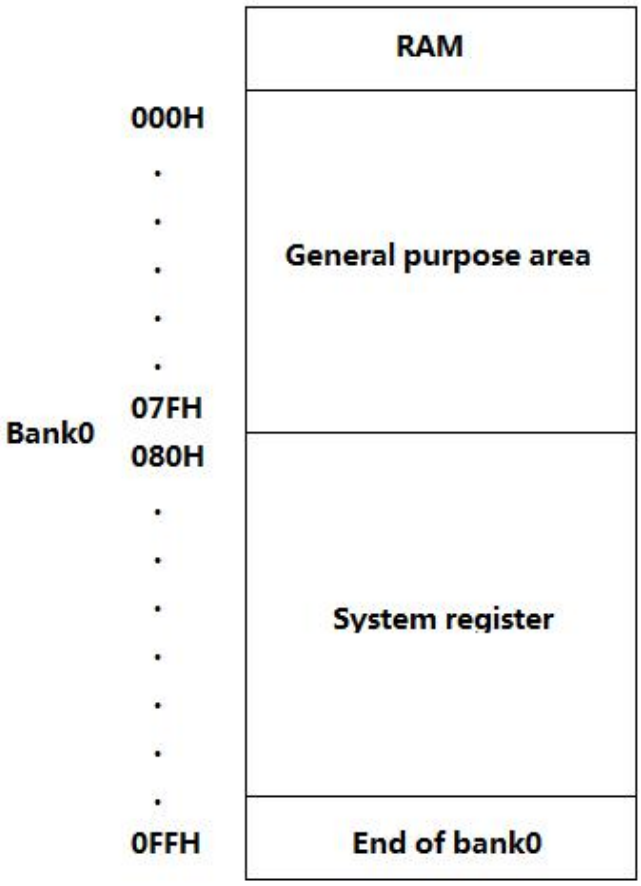
2 MEMORY STRUCTURE

2.1 PROGRAM MEMORY



2.2 DATA MEMORY

2.2.1 DATA REGISTER AREA (RAM)



2.2.2 SYSTEM REGISTER TABLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8			R	Z	Y		PFLAG	IRC CAL								
9																
A															P4CON	VREF
B		ADM	ADB	ADR	ADT				P0M							PEDGE
C					P4M	P5M			INTRQ	INTEN	OSCM		WDTR	TC0R	PCL	PCH
D	P0				P4	P5			T0M		TC0M	TC0C	TC1M	TC1C	TC1R	STKP
E	P0UR				P4UR	P5UR		@YZ								
F									STK3L	STK3H	STK2L	STK2H	STK1L	STK1H	STK0L	STK0H

2.2.3 SYSTEM REGISTER DESCRIPTION

R:
Working register and ROM look-up data buffer.

PFLAG:
ROM page and Special flag register.

VREF:
ADC reference voltage control register.

ADB:
ADC data buffer.

PEDGE:
P0.0 edge direction register.

INTRQ:
Interrupt request register.

OSCM:
Oscillator mode register.

TC0R:
TC0 auto-reload data buffer.

Pn:
Port n data buffer.

TC0M:
TC0 mode register.

TC1M:
TC1 mode register.

TC1R:
TC1 auto-reload data buffer.

PnUR:
Portn pull-up resister control register.

STK0～STK3:
Stack 0 ～stack 3 buffer.

IRCCAL:
IRC frequency adjustment register.

Y,Z：
Working, @YZ and ROM addressing register.

@YZ：
RAM YZ indirect addressing index pointer.

P4CON:
P4 configuration register.

ADM：
ADC mode register.

ADR：
ADC resolution select register.

ADT：
ADC offset calibration register.

PnM：
Port n input/output mode register.

INTEN:
Interrupt enable register.

WDTR:
Watchdog timer clear register.

PCH&L:
Program counter.

T0M：
T0 mode register.

TC0C:
TC0 counting register.

TC1C:
TC1 counting register.

STKP:
Stack pointer buffer.

2.2.4 BIT DEFINITION OF SYSTEM REGISTER

Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	R/W	Remarks
082H	RBIT7	RBIT6	RBIT5	RBIT4	RBIT3	RBIT2	RBIT1	RBIT0	R/W	R
083H	ZBIT7	ZBIT6	ZBIT5	ZBIT4	ZBIT3	ZBIT2	ZBIT1	ZBIT0	R/W	Z
084H	YBIT7	YBIT6	YBIT5	YBIT4	YBIT3	YBIT2	YBIT1	YBIT0	R/W	Y
086H	NT0	NPD	LVD36	LVD24		C	DC	Z	R/W	PFLAG
0AEH				P4CON4	P4CON3	P4CON2	P4CON1	P4CON0	R/W	P4CON
0AFH	EVHENB						VHS1	VHS0	R/W	VREF
0B1H	ADENB	ADS	EOC	GCHS		CHS2	CHS1	CHS0	R/W	ADM
0B2H	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	R	ADB
0B3H		ADCKS1		ADCKS0	ADB3	ADB2	ADB1	ADB0	R/W	ADR
0B4H	ADTS1	ADTS0		ADT4	ADT3	ADT2	ADT1	ADT0	R/W	ADT
0B8H		P06M	P05M		P03M	P02M	P01M	P00M	R/W	P0M
0BFH				P00G1	P00G0				R/W	PEDGE
0C4H				P44M	P43M	P42M	P41M	P40M	R/W	P4M
0C5H				P54M	P53M					
0C8H	ADCIRQ	TC1IRQ	TC0IRQ				P01IRQ	P00IRQ	R/W	INTRQ
0C9H	ADCIEN	TC1IEN	TC0IEN				P01IEN	P00IEN	R/W	INTEN
0CAH				CPUM1	CPUM0	CLKMD	STPHX		R/W	OSCM
0CCH	WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0	W	WDTR
0CDH	TC0R7	TC0R6	TC0R5	TC0R4	TC0R3	TC0R2	TC0R1	TC0R0	W	TC0R
0CEH	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	R/W	PCL
0CFH							PC9	PC8	R/W	PCH
0D0H		P06	P05	P04	P03	P02	P01	P00	R/W	P0
0D4H				P44	P43	P42	P41	P40	R/W	P4
0D5H				P54	P53				R/W	P5
0D8H					TC1X8	TC0X8	TC0GN		R/W	T0M
0DAH	TC0ENB	TC0rate	TC0rate	TC0rate	TC0CKS	ALOAD0	TC0OUT	PWM0OUT	R/W	TC0M
0DBH	TC0C7	TC0C6	TC0C5	TC0C4	TC0C3	TC0C2	TC0C1	TC0C0	R/W	TC0C
0DCH	TC1ENB	TC1rate	TC1rate	TC1rate	TC1CKS	ALOAD1	TC1OUT	PWM1OUT	R/W	TC1M
0DDH	TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0	R/W	TC1C
0DEH	TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0	W	TC1R
0DFH	GIE					STKPB2	STKPB1	STKPB0	R/W	STKP
0E0H		P06R	P05R		P03R	P02R	P01R	P00R	W	P0UR
0E4H				P44R	P43R	P42R	P41R	P40R	W	P4UR
0E5H				P54R	P53R				W	P5UR
0E7H	@YZ7	@YZ6	@YZ5	@YZ4	@YZ3	@YZ2	@YZ1	@YZ0	R/W	@YZ
0F8H	S3PC7	S3PC6	S3PC5	S3PC4	S3PC3	S3PC2	S3PC1	S3PC0	R/W	STK3L
0F9H							S3PC9	S3PC8	R/W	STK3H
0FAH	S2PC7	S2PC6	S2PC5	S2PC4	S2PC3	S2PC2	S2PC1	S2PC0	R/W	STK2L
0FBH							S2PC9	S2PC8	R/W	STK2H
0FCH	S1PC7	S1PC6	S1PC5	S1PC4	S1PC3	S1PC2	S1PC1	S1PC0	R/W	STK1L
0FDH							S1PC9	S1PC8	R/W	STK1H
0FEH	S0PC7	S0PC6	S0PC5	S0PC4	S0PC3	S0PC2	S0PC1	S0PC0	R/W	STK0L
0FFH							S0PC9	S0PC8	R/W	STK0H

3 FUNCTIONAL DESCRIPTION

3.1 SYSTEM REGISTER

3.1.1 082H-R (R REGISTER)

R register is an 8-bit buffer. There are two major functions of the register:

1. Can be used as working register;
2. For store high-byte data of look-up table;

(MOVC instruction executed, the high-byte data of specified ROM address will be stored in R register and the low-byte data will be stored in ACC)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RBIT7	RBIT6	RBIT5	RBIT4	RBIT3	RBIT2	RBIT1	RBIT0

3.1.2 083H-Z (Z REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ZBIT7	ZBIT6	ZBIT5	ZBIT4	ZBIT3	ZBIT2	ZBIT1	ZBIT0

3.1.3 084H-Y (Y REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
YBIT7	YBIT6	YBIT5	YBIT4	YBIT3	YBIT2	YBIT1	YBIT0

The Y and Z registers are the 8-bit buffers. There are three major functions of these registers:

1. Can be used as general working registers;
2. Can be used as RAM data pointers with @YZ register;
3. Can be used as ROM data pointer with the MOVC instruction for look-up table.

3.1.4 086H-PFLAG (PROGRAM FLAG)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NTO	NPD	LVD36	LVD24	--	C	DC	Z

Bit<7:6>: **NPO/NPD** - Reset status flag.

NTO	NPD	Reset Status	Description
0	0	Reset by WDT	Watch-dog time out
0	1	Reserved	-
1	0	Reset by LVD	VDD is lower than LVD detection value
1	1	Reset by external Reset Pin	The external reset pin detects low levels

Any reset situation requires a period of response time, and the system provides a complete reset process to ensure the smooth operation of reset. The time required to complete the reset varies for different types of oscillators. Therefore, the rising speed of VDD and the starting time of different crystal oscillations are not fixed. The RC oscillator has the shortest starting time and the crystal oscillator has a longer one. During the use of the MCU, attention should be paid to the reset time required by the product.

Bit<5>: **LVD36**- LVD 3.6V operating flag and only support LVD code option is LVD_H.
0:Inactive (VDD > 3.6V).
1: Active (VDD ≦ 3.6V).
Bit<4>: **LVD36**- LVD 2.4V operating flag and only support LVD code option is LVD_H.
0:Inactive (VDD > 2.4V).
1: Active (VDD ≦ 2.4V).

LVD	LVD Compiler Options		
	LVD_L	LVD_M	LVD_H
2.0V reset	Enable	Enable	Enable
2.4V flag	-	Enable	-
2.4V reset	-	-	Enable
3.6V flag	-	-	Enable

LVD_L: If VDD is less than 2.0V, the system is reset;
The flags of LVD24 and LVD36 are meaningless.
LVD_M: If VDD is less than 2.0V, the system is reset;
LVD24:If VDD > 2.4V, LVD24 =0; Otherwise, VDD <= 2.4V, LVD24=1;
LVD36: meaningless.
LVD_H: If VDD is less than 2.4V, the system is reset;
LVD36:If VDD > 3.6V, LVD36=0;Otherwise, VDD <= 3.6V, LVD36=1;

Note:
（1）When LVD reset ends, LVD24 and LVD36 will both be cleared;
（2）LVD 2.4V and LVD 3.6V detection level values are only used as design reference and cannot be used for accurate detection of chip operating voltage .

3.1.5 087H-IRCCAL（IRC FREQUENCY ADJUSTMENT REGISTER）

3.1.6 0AEH-P4CON（P4 CONFIGURATION REGISTER）

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0

Bit<4:0>:P4CON<4:0>-P4 ADC configuration control bits.
0:P4 can be an analog input (ADC input) or digital I/O pins.
1:P4 is pure analog input, can’t be a digital I/O pin.
Note: When Port 4.n is general I/O port not ADC channel, P4CON.n must be set to “0” or the Port 4.n digital I/O signal would be isolated.

3.1.7 0AFH-VREF（ADC REFERENCE VOLTAGE REGISTER）

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EVHENB	--	--	--	--	--	VHS1	VHS0

Bit<7>:**EVHENB**- ADC internal reference high voltage control bit.
0: Enable ADC internal VREFH function, P4.0/AIN0/VREFH pin is P4.0/AIN0 pin.
1: Disable ADC internal VREFH function, P4.0/AIN0/VREFH pin is external VREFH input pin.
Bit<1:0>:**VHS<1:0>**- ADC internal reference high voltage select bits.

VHS1	VHS0	Internal VREFH Voltage

0	0	2.0V
0	1	3.0V
1	0	4.0V
1	1	VDD

ADC builds in five high reference voltage source controlled through VREFH register. There are one external voltage source and four internal voltage source (VDD, 4V, 3V, 2V). When EVHENB bit is “1”, ADC reference voltage is external voltage source from P4.0. It is necessary to input a voltage to be ADC high reference voltage and not below 2V. If EVHENB bit is “0”, ADC reference voltage is from internal voltage source selected by VHS[1:0] bits. If VHS[1:0] is “11”,ADC reference voltage is VDD. If VHS[1:0] is “10”, ADC reference voltage is 4V. If VHS[1:0] is “01”, ADC reference voltage is 3V. If VHS[1:0] is “00”, ADC reference voltage is 2V. The limitation of internal reference voltage application is VDD can"t below each of internal voltage level, or the level is equal to VDD.

3.1.8 0B1H-ADM (ADC MODE REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADENB	ADS	EOC	GCHS	--	CHS2	CHS1	CHS0

Bit<7>:**ADENB**- ADC control bit. (In sleep mode, disable ADC to reduce power consumption)

0: Disable

1: Enable

Bit<6>:**ADS**-ADC start control bit. (ADS bit is cleared after ADC processing automatically)

0: Disable

1: Enable

Bit<5>:**EOC**-ADC status bit.

0: ADC progressing.

1: End of converting and reset ADS bit.

Bit<4>:**GCHS**-ADC global channel select bit.

0: Disable

1: Enable

Bit<2:0>: **CHS<2:0>**-ADC input channel select bit.

CHS<2:0>	CHANNEL	PIN
000	AIN0	P40
001	AIN1	P41
010	AIN2	P42
011	AIN3	P43
100	AIN4	P44
101	AIN5	Internal 1/4*VDD

The AIN5 is internal 1/4*VDD input channel. There is no any input pin from outside. AIN5 can be a good battery detector for battery system. To select appropriate internal VREFH level and compare value, a high performance and cheaper low battery detector is built in the system.

3.1.9 0B2H-ADB (ADC DATA BUFFER REGISTERS)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4

Bit<7:0>:**ADB<11:4>**-ADC data is stored in ADB (read only)

3.1.10 0B3H-ADR (ADC PRECISION SELECTION REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	ADCKS1	--	ADCKS0	ADB3	ADB2	ADB1	ADB0

Bit<6,4>:**ADCKS<1:0>**-ADC clock source selector bit.

ADCKS1	ADCKS0	ADC Clock Source
0	0	Fcpu/16
0	1	Fcpu/8
1	0	Fcpu
1	1	Fcpu/2

Bit<3:0>:**ADB<3:0>**- ADC data is stored in ADR.

3.1.11 0B4H-ADT (ADC ZERO DRIFT REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADTS1	ADTS0	--	ADT4	ADT3	ADT2	ADT1	ADT0

3.1.12 0B8H-P0M (P0 PORT MODE)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	P06M	P05M	P04M	P03M	P02M	P01M	P00M

Bit<6:0>:**P0M<6:0>**-P0 mode control bits.

0: input mode.

1: output mode.

NOTE: P0.4 input pin only, and the P0M.4 is undefined.

3.1.13 0BFH-PEDGE (EXTERNAL INTERRUPT OPERATION)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P00G1	P00G0	--	--	--

Bit<4:3>:**P00G<1:0>**-P00 edge trigger select bits.

00: reserved.

01: rising edge.

10: falling edge.

11: rising/falling bi-direction.

3.1.14 0C4H-P4M (P4 PORT MODE)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P44M	P43M	P42M	P41M	P40M

Bit<4:0>:**P4M<4:0>**-P4 mode control bits.

0: input mode.

1: output mode.

3.1.15 0C5H-P5M (P5 PORT MODE)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P54M	P53M	--	--	--

Bit<4:3>:**P5M<4:3>**-P5 mode control bits.

0: input mode.

1: output mode.

3.1.16 0C8H-INTRQ (INTRQ INTERRUPT REQUEST REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCIRQ	TC1IRQ	TC0IRQ	--	--	--	P01IRQ	P00IRQ

Bit<7>:**ADCIRQ**-ADC interrupt request flag.

Bit<6>:**TC1IEN**-TC1 timer interrupt request flag.

Bit<5>:**TC0IEN**-TC0 timer interrupt request flag.

Bit<1>:**P01IEN**-External P0.1 interrupt (INT1) request flag.

Bit<0>:**P00IEN**-External P0.0 interrupt (INT0) request flag.

0: None interrupt request.

1: Interrupt request.

3.1.17 0C9H-INTEN (INTEN INTERRUPT ENABLE REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCIEN	TC1IEN	TC0IEN	--	--	--	P01IEN	P00IEN

Bit<7>:**ADCIEN**- ADC interrupt control bit.

Bit<6>:**TC1IEN**- TC1 timer interrupt control bit.

Bit<5>:**TC0IEN**- TC0 timer interrupt control bit.

Bit<1>:**P01IEN**- External P0.1 interrupt (INT1) control bit.

Bit<0>:**P00IEN**- External P0.0 interrupt (INT0) control bit.

0: Disable

1: Enable

3.1.18 0CAH-OSCM (OSCM REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	CPUM1	CPUM0	CLKMD	STPHX	--

Bit<4:3>:**CPUM<1:0>**-CPU operating mode control bits.

CPUS1	CPUS0	Operating Mode
0	0	normal mode
0	1	sleep mode
1	0	green mode
1	1	reserved

Bit<2>:**CLKMD**- System high/Low clock mode control bit.

0: Normal (dual) mode. System clock is high clock.

1: Slow mode. System clock is internal low clock.

Bit<1>:**STPHX**- External high-speed oscillator control bit.

0: Free run.

1: Stop. Internal low-speed RC oscillator is still running.

3.1.19 0CCH-WDTR (WATCHDOG CLEAR REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0

Watchdog clear is controlled by WDTR register. Moving 0x5A data into WDTR is to reset watchdog timer.

3.1.20 0CDH-TC0R (TC0R AUTO-LOAD REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0R7	TC0R6	TC0R5	TC0R4	TC0R3	TC0R2	TC0R1	TC0R0

The equation of TC0R initial value is as following:

TC0R initial value = N- (TC0 interrupt interval time * input clock)

N is TC0 overflow boundary number. TC0 overflow time and valid value as follow table:

TC0CKS	TC0X8	PWM0	ALOAD0	TC0OUT	N	TC0R Value
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

Example: To set 10ms interval time for TC0 interrupt. TC0 clock source is Fcpu (TC0KS=0, TC0X8=0) and no PWM output (PWM0=0). High clock is external 4MHz. Fcpu=Fhosc/4. Select TC0RATE=010 (Fcpu/64).

TC0R initial value = N - (TC0 interrupt interval time * input clock)
= 256 - (10ms * 4MHz / 4 / 64)
= 256 - (0.01* 4000000/ 4 / 64)
= 100 = 64H

3.1.21 0CEH-PCL (PROGRAM COUNTER LOW-BYTE)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0

3.1.22 0CFH-PCH (PROGRAM COUNTER HIGH-BYTE)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	--	--	--	PC9	PC8

The program counter (PC) is a 10-bit binary counter separated into the high-byte 2 and the low-byte 8 bits. This counter is responsible for pointing a location in order to fetch an instruction for kernel circuit. Normally, the program counter is automatically incremented with each instruction during program execution.

Besides, it can be replaced with specific address by executing CALL or JMP instruction. When JMP or CALL instruction is executed, the destination address will be inserted to bit 0 ~ bit 9.

3.1.23 0D0H-P0 (P0 PORT DATA REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	P06	P05	P04	P03	P02	P01	P00

Bit<6:0>:**P0<6:0>**-P0 PORT DATA REGISTER

Note: The P04 keeps "1" when external reset enable by code option.

3.1.24 0D4H-P4 (P4 PORT DATA REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P44	P43	P42	P41	P40

Bit<4:0>:**P4<4:0>**-P4 PORT DATA REGISTER.

3.1.25 0D5H-P5 (P5 PORT DATA REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P54	P53	--	--	--

Bit<4:3>:**P5<4:3>**-P5 PORT DATA REGISTER.

3.1.26 0D8H-T0M (TC0X8, TC0GN FLAGS)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	--	TC1X8	TC0X8	TC0GN	--

Bit<3>:**TC1X8**- TC1 internal clock source control bit.

0:TC1 internal clock source is Fcpu. TC1RATE is from Fcpu/2~Fcpu/256.

1:TC1 internal clock source is Fosc. TC1RATE is from Fosc/1~Fosc/128.

Note: Under TC1 event counter mode (TC1CKS=1), TC1X8 bit and TC1RATE are useless.

Bit<2>:**TC0X8**- TC0 internal clock source control bit.

0:TC0 internal clock source is Fcpu. TC0RATE is from Fcpu/2~Fcpu/256.

1:TC0 internal clock source is Fosc. TC0RATE is from Fosc/1~Fosc/128.

Bit<1>:**TC0GN**- TC0 green mode wake-up function control bit.

0: Disable

1: Enable

Note: Under TC0 event counter mode (TC0CKS=1), TC0X8 bit and TC0RATE are useless.

3.1.27 0DAH-TC0M (TC0M MODE REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0ENB	TC0rate2	TC0rate1	TC0rate0	TC0CKS	ALOAD0	TC0OUT	PWM0OUT

Bit<7>:**TC0ENB**- TC0 counter control bit.

0: Disable

1: Enable

Bit<6:4>:**TC0rate<2:0>**:TC0 internal clock select bits.

TC0rate<2:0>	TC0X8=0	TC0X8=1
000	Fcpu/256	Fosc/128
001	Fcpu/128	Fosc/64
010	Fcpu/64	Fosc/32

011	Fcpu/32	Fosc/16
100	Fcpu/16	Fosc/8
101	Fcpu/8	Fosc/4
110	Fcpu/4	Fosc/2
111	Fcpu/2	Fosc/1

Bit<3>:**TC0CKS**-TC0 clock source select bit.
0: Internal clock (Fcpu and Fhosc controlled by TC0X8 bit).
1: External input pin (P0.0/INT0) and enable event counter function.
Bit<2>:**ALOAD0**- Auto-reload control bit. Only valid when PWM0OUT = 0.
0: Disable
1: Enable
Bit<1>:**TC0OUT**- TC0 time out toggle signal output control bit.
(Only valid when PWM0OUT = 0.)
0:Disable, P5.4 is GPIO mode.
1:Enable, P5.4 outputs TC0 signal.
Bit<0>:**PWM0OUT**- PWM output control bit.
0: Disable
1: Enable

Note: When TC0CKS=1, TC0 became an external event counter and TC0RATE is useless. No more P0.0 interrupt request will be raised. (P0.0IRQ will be always 0).

3.1.28 0DBH-TC0C (TC0C COUNTING REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0C7	TC0C6	TC0C5	TC0C4	TC0C3	TC0C2	TC0C1	TC0C0

The equation of TC0C initial value is as following:

TC0C initial value = N - (TC0 interrupt interval time * input clock)

N is TC0 overflow boundary number. The setting of parameters in each mode is shown in the following table:

TC0CKS	TC0X8	PWM0	ALOAD0	TC0OUT	N	TC0C Value
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

3.1.29 0DCH-TC1M (TC1M MODE REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1ENB	TC1rate2	TC1rate1	TC1rate0	TC1CKS	ALOAD1	TC1OUT	PWM1OUT

Bit<7>:**TC1ENB**- TC1 counter control bit.

0:Disable

1:Enable

Bit<6:4>:**TC1rate<2:0>**: TC1 internal clock select bits.

TC1rate<2:0>	TC1X8=0	TC1X8=1
000	Fcpu/256	Fosc/128
001	Fcpu/128	Fosc/64
010	Fcpu/64	Fosc/32
011	Fcpu/32	Fosc/16
100	Fcpu/16	Fosc/8
101	Fcpu/8	Fosc/4
110	Fcpu/4	Fosc/2
111	Fcpu/2	Fosc/1

Bit<3>:**TC1CKS**- TC1 clock source select bit.

0:Internal clock (Fcpu and Fhosc controlled by TC1X8 bit).

1:External input pin (P0.1/INT1) and enable event counter function, TC1rate[2:0] bits are useless.

Bit<2>:**ALOAD1**- Auto-reload control bit. Only valid when PWM1OUT = 0.

0: Disable

1: Enable

Bit<1>:**TC1OUT**- TC1 time out toggle signal output control bit.

(Only valid when PWM1OUT = 0.)

0:Disable, P5.3 is GPIO mode.

1:Enable, P5.3 outputs TC1/2 buzzer signal.

Bit<0>:**PWM1OUT**- PWM output control bit.

0:Disable, P5.3 is GPIO mode.

1:Enable, P5.3 outputs PWM signal. PWM duty controlled by TC1OUT, ALOAD1 bits.

Note: When TC1CKS=1, TC1 became an external event counter and TC1RATE is useless. No more P0.1 interrupt request will be raised. (P0.1IRQ will be always 0).

3.1.30 0DDH-TC1C (TC1C COUNTING REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0

The equation of TC1C initial value is as following:

TC1C initial value = N - (TC1 interrupt interval time * input clock)

N is TC1 overflow boundary number. The setting of parameters in each mode is shown in the following table:

TC1CKS	TC1X8	PWM1	ALOAD1	TC1OUT	N	TC1C Value
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

3.1.31 0DEH-TC1R (TC1R AUTO-LOAD REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0

The equation of TC1R initial value is as following:

TC1R initial value = N - (TC1 interrupt interval time * input clock)

N is TC1 overflow boundary number. TC1 overflow time and valid value as follow table:

TC1CKS	TC1X8	PWM1	ALOAD1	TC1OUT	N	TC1R Value
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

Example: To set 10ms interval time for TC1 interrupt. TC1 clock source is Fcpu (TC1KS=0, TC1X8=0) and no PWM output (PWM1=0). High clock is external 4MHz. Fcpu=Fhosc/4. Select TC1RATE=010 (Fcpu/64).

TC1R = N - (TC1 interrupt interval time * input clock)
= 256 - (10ms * 4MHz / 4 / 64)
= 256 - (0.01* 4000000/ 4 / 64)
= 100 = 64H

3.1.32 0DFH-STKP (STACK REGISTERS)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GIE	--	--	--	--	STKPB2	STKPB1	STKPB0

Bit<7>:**GIE**- Global interrupt control bit.

0: Disable

1:Enable

Bit<2:0>:**STKPB<2:0>**: Stack pointer.

Example: Stack pointer (STKP) reset, we strongly recommended to clear the stack pointers in the beginning of the program:

```
MOV     A,#00000111B
BOMOV   STKP,A
```

3.1.33 0E0H-P0UR (P0 PULL UP CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	P06R	P05R	--	P03R	P02R	P01R	P00R

Bit<6:0>:**P0UP<6:0>**-P0 pull up control bit.

0:Disable

1:Enable

Note: P0.4 is input only pin and without pull-up resister. The P0UR.4 is undefined.

3.1.34 0E4H-P4UR (P4 PULL UP CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P44R	P43R	P42R	P41R	P40R

Bit<4:0>:**P4UR<4:0>**-P4 pull up control bit.

0:Disable

1:Enable

3.1.35 0E5H-P5UR (P5 PULL UP CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	P54R	P53R	--	--	--

Bit<4:3>:**P5DR<4:3>**-P5 pull up control bit.

0:Disable

1:Enable

3.1.36 0E7H-@YZ (@YZ REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
@YZ7	@YZ6	@YZ5	@YZ4	@YZ3	@YZ2	@YZ1	@YZ0

The indirectly addressing mode is to access the memory by the data pointer registers (Y/Z).

Example: Indirectly addressing mode with @YZ register.

```
BOMOV     Y,#0        ; To clear Y register to access RAM bank 0.
BOMOV     Z,#12H      ; To set an immediate data 12H into Z register.
BOMOV     A,@YZ
```

3.1.37 0F8H～0FFH-STK0～STK3 (STACK REGISTERS)

3.2 GPIO PORT

JZ8P2612 has three sets of bidirectional I/O ports, 14 inputs and 13 outputs, most of which can be reused for other functions.

13 programmable pull up I/O pins: P00～P03, P05～P06, P40～P44, P53～P54;

Port characteristics are as follows:

Port	Flip feature
P00-P01, P05-P06	0.8VDD/0.3VDD
P02-P03	0.7VDD/0.4VDD
P04	0.7VDD/0.2VDD
P40-P44	0.8VDD/0.3VDD
P53-P54	0.8VDD/0.3VDD

3.2.1 GPIO REGISTER SPECIFICATION

0B8H (P0M PORT MODE)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	P06M	P05M	P04M	P03M	P02M	P01M	P00M

Bit<6:0>:**P0M<6:0>**- P0 mode control bits.

0: input mode.

1: output mode.

NOTE: P0.4 input pin only, and the P0M.4 is undefined.

0C4H (P4M PORT MODE)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P44M	P43M	P42M	P41M	P40M

Bit<4:0>:**P4M<4:0>**-P4 mode control bits.

0: input mode.

1: output mode.

0C5H (P5M PORT MODE)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P54M	P53M	-	-	-

Bit<4:3>:**P5M<4:3>**-P5 mode control bits.

0: input mode.

1: output mode.

0D0H (P0 PORT DATA REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	P06	P05	-	P03	P02	P01	P00

Bit<6:0>:**P0<6:0>**-P0 PORT DATA REGISTER

Note: The P04 keeps “1” when external reset enable by code option.

0D4H (P4 PORT DATA REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P44	P43	P42	P41	P40

Bit<4:0>:**P4<4:0>**-P4 PORT DATA REGISTER.

0D5H (P5 PORT DATA REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P54	P53	-	-	-

Bit<4:3>:**P5<4:3>**-P5 PORT DATA REGISTER.

0E0H (P0UR PULL UP CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	P06R	P05R	-	P03R	P02R	P01R	P00R

Bit<6:0>:**P0UP<6:0>**-P0 pull up control bit.

0:Disable

1:Enable

Note: P0.4 is input only pin and without pull-up resister. The P0UR.4 is undefined.

0E4H (P4UR PULL UP CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P44R	P43R	P42R	P41R	P40R

Bit<4:0>:**P4UR<4:0>**-P4 pull up control bit.

0:Disable

1:Enable

0E5H (P5UR PULL UP CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P54R	P53R	-	-	-

Bit<4:3>:**P5DR<4:3>**-P5 pull up control bit.

0:Disable

1:Enable

3.3 TIMERS

3.3.1 WATCHDOG TIMER

The watchdog timer (WDT) is a binary up counter designed for monitoring program execution. If the program goes into the unknown status by noise interference, WDT overflow signal raises and resets MCU. Watchdog clock controlled by code option and the clock source is internal low-speed RC oscillator（16KHz @3V, 32KHz @5V）.

Watchdog overflow time = 8192 / Internal Low-Speed oscillator (sec).

VDD	Internal Low RC	Watchdog Overflow Time
3V	16KHz	512ms
5V	32KHz	256ms

The watchdog timer has three operating options controlled “WatchDog” code option:

Disable: Disable watchdog timer function.

Enable: Enable watchdog timer function. Watchdog timer actives in normal mode and slow mode. In sleep mode and green mode, the watchdog timer stops.

Always_On: Enable watchdog timer function. The watchdog timer actives and not stop in sleep mode and green mode.

In high noisy environment, the “Always_On” option of watchdog operations is the strongly recommendation to make the system reset under error situations and re-start again.

Watchdog clear is controlled by WDTR register. Moving 0x5A data into WDTR is to reset watchdog timer.

OCCH (WDTR WATCHDOG CLEAR REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDT0

Example: An operation of watchdog timer is as following. To clear the watchdog timer counter in the top of the main routine of the program.

```
MOV     A,#5AH          ; Clear the watchdog timer.
BOMOV   WDTR,A
```

Watchdog timer application note is as following:

(1) Before clearing watchdog timer, check I/O status and check RAM contents can improve system error;

(2) Don"t clear watchdog timer in interrupt vector and interrupt service routine. That can improve main routine fail;

(3) Clearing watchdog timer program is only at one part of the program. This way is the best structure to enhance the watchdog timer function;

3.3.2 TIMER/COUNTER (TC0)

3.3.2.1 OVERVIEW

The TC0 timer is an 8-bit binary up timer with basic timer, event counter, PWM and buzzer functions. The basic timer function supports flag indicator (TC0IRQ bit) and interrupt operation (interrupt vector). The interval time is programmable through TC0M, TC0C, TC0R registers. The event counter is changing TC0 clock source from system

clock (Fcpu/Fhosc) to external clock like signal (e.g. continuous pulse, R/C type oscillating signal…). TC0 becomes a counter to count external clock number to implement measure application. TC0 builds in duty/cycle programmable PWM. The PWM cycle and resolution are controlled by TC0M and TC0R registers. TC0 builds in buzzer function to output TC0/2 signal. TC0 supports auto-reload function. When TC0 timer overflow occurs, the TC0C will be reloaded from TC0R automatically. The TC0 builds in green mode wake-up function controlled by TC0GN bit.

The main purposes of the TC0 timer are as following:

(1)8-bit programmable up counting timer: Generate time-out at specific time intervals based on the selected clock frequency.

(2)Interrupt function: TC0 timer function supports interrupt function. When TC0 timer occurs overflow, the TC0IRQ actives and the system points program counter to interrupt vector to do interrupt sequence.

(3)Event Counter: The event counter function counts the external clock counts.

(4)PWM output: The PWM is duty/cycle programmable controlled by TC0rate, TC0R, ALOAD0 and TC0OUT registers.

(5)Buzzer output: The Buzzer output signal is 1/2 cycle of TC0 interval time.

(6)Green mode function: All TC0 functions (timer, PWM, Buzzer, event counter, auto-reload) keeps running in green mode. TC0 builds in green mode wake-up function when TC0 overflow occurs controlled by TC0GN bit.

3.3.2.2 TC0 TIMER OPERATION

TC0 timer is controlled by TC0ENB bit. When TC0ENB=0, TC0 timer stops. When TC0ENB=1, TC0 timer starts to count. Before enabling TC0 timer, setup TC0 timer" s configurations to select timer function modes, e.g. basic timer, interrupt function…TC0C increases “1” by timer clock source. When TC0 overflow event occurs, TC0IRQ flag is set as ”1” to indicate overflow and cleared by program. The overflow condition is TC0C count from full scale (0xFF) to zero scale (0x00). In difference function modes, TC0C value relates to operation. If TC0C value changing effects operation, the transition of operations would make timer function error. So TC0 builds in double buffer to avoid these situations happen. The double buffer concept is to flash TC0C during TC0 counting, to set the new value to TC0R (reload buffer), and the new value will be loaded from TC0R to TC0C after TC0 overflow occurrence automatically. In the next cycle, the TC0 timer runs under new conditions, and no any transitions occur. The auto-reload function is controlled by ALOAD0 bit in timer/counter mode, and enabled automatically in PWM mode as TC0 enables. If TC0 timer interrupt function is enabled (TC0IEN=1), the system will execute interrupt procedure. The interrupt procedure is system program counter points to interrupt vector (ORG 8) and executes interrupt service routine after TC0 overflow occurrence. Clear TC0IRQ by program is necessary in interrupt procedure. TC0 timer can works in normal mode,

slow mode and green mode. But in green mode, TC0 keep counting, set TC0IRQ and outputs PWM and buzzer, and wake-up system controlled by TC0GN bit.

TC0 provides different clock sources to implement different applications and configurations. TC0 clock source includes Fcpu (instruction cycle), Fhosc (high speed oscillator) and external input pin (P0.0) controlled by TC0CKS and TC0X8 bits. TC0X8 bit selects the clock source is from Fcpu or Fhosc. If TC0X8=0, TC0 clock source is Fcpu through TC0rate[2:0] pre-scalar to decide Fcpu/2~Fcpu/256. If TC0X8=1, TC0 clock source is Fhosc through TC0rate[2:0] pre-scalar to decide Fhosc/1～Fhosc/128. TC0CKS bit controls the clock source is external input pin or controlled by TC0X8 bit. If TC0CKS=0, TC0 clock source is selected by TC0X8 bit. If TC0CKS=1, TC0 clock source is external input pin that means to enable event counter function. TC0rate[2:0] pre-scalar is unless when TC0CKS=1 condition. TC0 length is 8-bit (256 steps), and the one count period is each cycle of input clock.

TC0CKS	TC0Rate[2:0]	TC0 clock	TC0 Interval Time			
			Fhosc=16MHz Fcpu=Fhosc/4		Fhosc=4MHz Fcpu=Fhosc/4	
1			max. (ms)	Unit (us)	max. (ms)	Unit (us)
0	000b	Fcpu/256	16.384	64	65.536	256
0	001b	Fcpu/128	8.192	32	32.768	128
0	010b	Fcpu/64	4.096	16	16.384	64
0	011b	Fcpu/32	2.048	8	8.192	32
0	100b	Fcpu/16	1.024	4	4.096	16
0	101b	Fcpu/8	0.512	2	2.048	8
0	110b	Fcpu/4	0.256	1	1.024	4
0	111b	Fcpu/2	0.128	0.5	0.512	2
1	000b	Fhosc/128	2.048	8	8.192	32
1	001b	Fhosc/64	1.024	4	4.096	16
1	010b	Fhosc/32	0.512	2	2.048	8
1	011b	Fhosc/16	0.256	1	1.024	4
1	100b	Fhosc/8	0.128	0.5	0.512	2
1	101b	Fhosc/4	0.064	0.25	0.256	1
1	110b	Fhosc/2	0.032	0.125	0.128	0.5
1	111b	Fhosc/1	0.016	0.0625	0.064	0.25

3.3.2.3 TCOM MODE REGISTER

TCOM is TC0 timer mode control register to configure TC0 operating mode including TC0 pre-scaler, clock source, PWM function…These configurations must be setup completely before enabling TC0 timer.

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0ENB	TC0rate2	TC0rate1	TC0rate0	TC0CKS	ALOAD0	TC0OUT	PWM0OUT

Bit<7>:**TC0ENB**-TC0 counter control bit.
0:Disable
1:Enable

Bit<6:4>:**TC0rate<2:0>**:TC0 internal clock select bits.

TC0rate<2:0>	TC0X8=0	TC0X8=1
000	Fcpu/256	Fosc/128
001	Fcpu/128	Fosc/64
010	Fcpu/64	Fosc/32
011	Fcpu/32	Fosc/16
100	Fcpu/16	Fosc/8
101	Fcpu/8	Fosc/4
110	Fcpu/4	Fosc/2
111	Fcpu/2	Fosc/1

Bit<3>:**TC0CKS**-TC0 clock source select bit.
0:Internal clock (Fcpu and Fhosc controlled by TC0X8 bit).
1: External input pin (P0.0/INT0)

Bit<2>:**ALOAD0**- Auto-reload control bit. Only valid when PWM0OUT = 0.
0:Disable
1:Enable

Bit<1>:**TC0OUT**- TC0 time out toggle signal output control bit. Only valid when PWM0OUT = 0.
0: Disable,P5.4 is GPIO mode.
1: Enable,P5.4 outputs TC0OUT signal.

Bit<0>:**PWM0OUT**-PWM output control bit.
0: Disable
1: Enable, PWM duty controlled by TC0OUT,ALOAD0 bits.

Note: When TC0CKS=1, TC0 became an external event counter and TC0RATE is useless. No more P0.0 interrupt request will be raised. (P00IRQ=0)

3.3.2.4 TC0X8/TC0GN FLAGS

TC0 clock source selection and green mode wake-up function are controlled by T0M.

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	--	TC1X8	TC0X8	TC0GN	--

Bit<2>:**TC0X8**- TC0 internal clock source control bit.
0: TC0 internal clock source is Fcpu. TC0RATE is from Fcpu/2~Fcpu/256.
1: TC0 internal clock source is Fhosc. TC0RATE is from Fhosc/1~Fhosc/128.

Bit<1>:**TC0GN**- TC0 green mode wake-up function control bit.

0: Disable
1: Enable

Note: Under TC0 event counter mode (TC0CKS=1), TC0X8 bit and TC0RATE are useless.

3.3.2.5 TC0C COUNTING REGISTER

TC0C is TC0 8-bit counter. When TC0C overflow occurs, the TC0IRQ flag is set as "1" and cleared by program. The TC0C decides TC0 interval time through below equation to calculate a correct value. It is necessary to write the correct value to TC0C register and TC0R register, and then enable TC0 timer. After TC0 overflow occurs, the TC0C register is loaded a correct value from TC0R register automatically, not program.

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0C7	TC0C6	TC0C5	TC0C4	TC0C3	TC0C2	TC0C1	TC0C0

The equation of TC0C initial value is as following:

TC0C initial value = N - (TC0 interrupt interval time * input clock)

N is TC0 overflow boundary number. The setting of parameters in each mode is shown in the following table:

TC0CKS	TC0X8	PWM0	ALOAD0	TC0OUT	N	TC0C Value
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

3.3.2.6 TC0R AUTO-LOAD REGISTER

TC0 timer builds in auto-reload function, and TC0R register stores reload data. When TC0C overflow occurs, TC0C register is loaded data from TC0R register automatically. Under TC0 timer counting status, to modify TC0 interval time is to modify TC0R register, not TC0C register. New TC0C data of TC0 interval time will be updated after TC0 timer overflow occurrence, TC0R loads new value to TC0C register. But at the first time to setup TC0M, TC0C and TC0R must be set the same value before enabling TC0 timer. TC0 is double buffer design. If new TC0R value is set by program, the new value is stored in 1st buffer. Until TC0 overflow occurs, the new value moves to real TC0R buffer. This way can avoid any transitional condition to effect the correctness of TC0 interval time and PWM output signal.

Note: Under PWM mode, auto-load is enabled automatically. The ALOAD0 bit is selecting overflow boundary.

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0R7	TC0R6	TC0R5	TC0R4	TC0R3	TC0R2	TC0R1	TC0R0

The equation of TC0R initial value is as following:

TC0R initial value = N-（TC0 interrupt interval time * input clock）

N is TC0 overflow boundary number. TC0 overflow time and valid value as follow table:

TC0CKS	TC0X8	PWM0	ALOAD0	TC0OUT	N	TC0R Value
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

Example: To set 10ms interval time for TC0 interrupt. TC0 clock source is Fcpu (TC0KS=0, TC0X8=0) and no PWM output (PWM0=0). High clock is external 4MHz. Fcpu=Fhosc/4. Select TC0RATE=010 (Fcpu/64).

TC0R initial value = N - (TC0 interrupt interval time * input clock)
= 256 -（10ms * 4MHz / 4 / 64）
= 256 -（0.01* 4000000/ 4 / 64）
= 100 = 64H

3.3.2.7 TC0 EVENT COUNTER

TC0 event counter is set the TC0 clock source from external input pin (P0.0). When TC0CKS=1, TC0 clock source is switch to external input pin (P0.0). TC0 event counter trigger direction is falling edge. When one falling edge occurs, TC0C will up one count. When TC0C counts from 0xFF to 0x00, TC0 triggers overflow event. The external event counter input pin"s wake-up function of GPIO mode is disabled when TC0 event counter function enabled to avoid event counter signal trigger system wake-up and not keep in power saving mode. The external event counter input pin" s external interrupt function is also disabled when TC0 event counter function enabled, and the P00IRQ bit keeps “0” status. The event counter usually is used to measure external continuous signal rate, e.g. continuous pulse, R/C type oscillating signal.These signal phase don"t synchronize with MCU’ s main clock. Use TC0 event to measure it and calculate the signal rate in program for different applications.

3.3.2.8 TC0 CLOCK FREQUENCY OUTPUT

Buzzer output (TC0OUT) is from TC0 timer/counter frequency output function. By setting the TC0 clock frequency, the clock signal is output to P5.4 and the P5.4 general purpose I/O function is auto-disable. The TC0OUT frequency is divided by 2 from TC0 interval time. TC0OUT frequency is 1/2 TC0 frequency. The TC0 clock has many combinations and easily to make difference frequency.

When buzzer outputs, TC0IRQ still actives as TC0 overflows, and TC0 interrupt function actives as TC0IEN = 1. But strongly recommend be careful to use buzzer and TC0 timer together, and make sure both functions work well. The buzzer output pin is shared with GPIO and switch to output buzzer signal as TC0OUT=1 automatically. If TC0OUT bit is cleared to disable buzzer signal, the output pin exchanges to last GPIO mode automatically.

Example: Setup TC0OUT output from TC0 to TC0OUT (P5.4). The external high-speed clock Fhosc is 4MHz. the instruction cycle Fcpu is Fhosc/4. The TC0OUT frequency is 0.5KHz. Because the TC0OUT signal is divided by 2, set the TC0 clock to 1KHz. The TC0 clock source is from external oscillator clock. TC0 rate is Fcpu/8. The TC0RATE2～TC0RATE1 = 101. TC0C = TC0R = 131.

```
MOV     A,#01010000B
BOMOV   TC0M,A      ; Set the TC0 rate to Fcpu/8
MOV     A,#131      ; Set the auto-reload reference value
BOMOV   TC0C,A
BOMOV   TC0R,A
BOBSET FTC0OUT     ; Enable TC0 output to P5.4 and disable P5.4 I/O function
BOBSET FALOAD0     ; Enable TC0 auto-reload function
BOBSET FTC0ENB     ; Enable TC0 timer
```

Note: Buzzer output is enable, and “PWM0OUT” must be “0”.

3.3.2.9 PULSE WIDTH MODULATION(PWM)

The PWM is duty/cycle programmable design to offer various PWM signals. When TC0 timer enables and PWM0OUT bit sets as “1” (enable PWM outputs), the PWM output pin outputs PWM signal. One cycle of PWM signal is high pulse first, and then low pulse outputs. TC0RATE, ALOAD0, TC0OUT bits control the cycle of PWM, and TC0R decides the duty (high pulse width length) of PWM. TC0C initial value must be set to zero when TC0 timer enables. When TC0C count is equal to TC0R, the PWM high pulse finishes and exchanges to low level. When TC0 overflows, one complete PWM cycle finishes. The PWM exchanges to high level for next cycle. If modify the PWM cycle by program as PWM outputting, the new cycle occurs at next cycle when TC0C loaded from TC0R.

The PWM builds in four programmable resolution (1/256, 4/64, 4/32, 4/16) controlled by ALOAD0 and TC0OUT bits as PWM0OUT = 1.

PWM0	ALOAD0	TC0OUT	PWM Resolution	TC0R valid value	TC0R value
1	0	0	256	00H~0FFH	00000000B~11111111B
1	0	1	64	00H~3FH	xx000000B~xx111111B
1	1	0	32	00H~1FH	xxx00000B~xxx11111B
1	1	1	16	00H~0FH	xxxx0000B~xxxx1111B

TC0R controls the high pulse width of PWM for PWM’s duty. When TC0C = TC0R, PWM output exchanges to low status. When PWM outputs, TC0IRQ still actives as TC0 overflows, and TC0 interrupt function actives as TC0IEN = 1. TC0 interrupt interval time is equal to PWM’s cycle in PWM mode. That means TC0 interrupt period has four

resolution following ALOAL0 and TC0OUT values. But strongly recommend be careful to use PWM and TC0 timer together, and make sure both functions work well.

The PWM output pin is shared with GPIO and switch to output PWM signal as PWM0OUT=1 automatically. If PWM0OUT bit is cleared to disable PWM, the output pin exchanges to last GPIO mode automatically. It easily to implement carry signal on/off operation, not to control TC0ENB bit.

3.3.3 TIMER/COUNTER 1 (TC1)

3.3.3.1 OVERVIEW

The TC1 timer is an 8-bit binary up timer with basic timer, event counter, PWM and buzzer functions. The basic timer function supports flag indicator (TC1IRQ bit) and interrupt operation (interrupt vector). The interval time is programmable through TC1M, TC1C, TC1R registers. The event counter is changing TC1 clock source from system clock (Fcpu/Fhosc) to external clock like signal (e.g. continuous pulse, R/C type oscillating signal). TC1 becomes a counter to count external clock number to implement measure application. TC1 builds in duty/cycle programmable PWM. The PWM cycle and resolution are controlled by TC1M and TC1R registers. TC1 builds in buzzer function to output TC1/2 signal. TC1 supports auto-reload function. When TC1 timer overflow occurs, the TC1C will be reloaded from TC1R automatically.

The main purposes of the TC1 timer are as following.

(1) 8-bit programmable up counting timer: Generate time-out at specific time intervals based on the selected clock frequency.

(2)Interrupt function: TC1 timer function supports interrupt function. When TC1 timer occurs overflow, the TC1IRQ actives and the system points program counter to interrupt vector to do interrupt sequence.

(3)Event Counter: The event counter function counts the external clock counts.

(4) PWM output: The PWM is duty/cycle programmable controlled by TC1rate, TC1R, ALOAD1 and TC1OUT registers.

(5)Buzzer output: The Buzzer output signal is 1/2 cycle of TC1 interval time.

(6) Green mode function: All TC1 functions (timer, PWM, Buzzer, event counter, auto-reload) keeps running in green mode and no wake-up function.

3.3.3.2 TC1 TIMER OPERATION

TC1 timer is controlled by TC1ENB bit. When TC1ENB=0, TC1 timer stops. When TC1ENB=1, TC1 timer starts to count. Before enabling TC1 timer, setup TC1 timer's configurations to select timer function modes, e.g. basic timer, interrupt function. TC1C increases "1" by timer clock source. When TC1 overflow event occurs, TC1IRQ flag is set as "1" to indicate overflow and cleared by program. The overflow condition is TC1C count from full scale (0xFF) to zero scale (0x00). In difference function modes, TC1C value relates to operation. If TC1C value changing effects operation, the transition of operations would make timer function error. So TC1 builds in double buffer to avoid these situations happen. The double buffer concept is to flash TC1C during TC1 counting, to set the new value to TC1R (reload buffer), and the new value will be loaded from TC1R to TC1C after TC1 overflow occurrence automatically. In the next cycle, the TC1 timer runs under new conditions, and no any transitions occur. The auto-reload function is controlled by ALOAD1 bit in

timer/counter mode, and enabled automatically in PWM mode as TC1 enables. If TC1 timer interrupt function is enabled (TC1IEN=1), the system will execute interrupt procedure. The interrupt procedure is system program counter points to interrupt vector (ORG 8) and executes interrupt service routine after TC1 overflow occurrence. Clear TC1IRQ by program is necessary in interrupt procedure. TC1 timer can works in normal mode, slow mode and green mode. But in green mode, TC1 keep counting, set TC1IRQ and outputs PWM and buzzer, but can't wake-up system.

TC1 provides different clock sources to implement different applications and configurations. TC1 clock source includes Fcpu (instruction cycle), Fhosc (high speed oscillator) and external input pin (P0.1) controlled by TC1CKS and TC1X8 bits. TC1X8 bit selects the clock source is from Fcpu or Fhosc. If TC1X8=0, TC1 clock source is Fcpu through TC1rate[2:0] pre-scalar to decide Fcpu/2～Fcpu/256. If TC1X8=1, TC1 clock source is Fhosc through TC1rate[2:0] pre-scalar to decide Fhosc/1～Fhosc/128. TC1CKS bit controls the clock source is external input pin or controlled by TC1X8 bit. If TC1CKS=0, TC1 clock source is selected by TC1X8 bit. If TC1CKS=1, TC1 clock source is external input pin that means to enable event counter function. TC1rate[2:0] pre-scalar is unless when TC1CKS=1 condition.

TC1X8	TC1Rate[2:0]	TC1 Clock	TC1 Interval Time			
			Fhosc=16MHz Fcpu=Fhosc/4		Fhosc=4MHz Fcpu=Fhosc/4	
			max. (ms)	Unit (us)	max. (ms)	Unit (us)
0	000b	Fcpu/256	16.384	64	65.536	256
0	001b	Fcpu/128	8.192	32	32.768	128
0	010b	Fcpu/64	4.096	16	16.384	64
0	011b	Fcpu/32	2.048	8	8.192	32
0	100b	Fcpu/16	1.024	4	4.096	16
0	101b	Fcpu/8	0.512	2	2.048	8
0	110b	Fcpu/4	0.256	1	1.024	4
0	111b	Fcpu/2	0.128	0.5	0.512	2
1	000b	Fhosc/128	2.048	8	8.192	32
1	001b	Fhosc/64	1.024	4	4.096	16
1	010b	Fhosc/32	0.512	2	2.048	8
1	011b	Fhosc/16	0.256	1	1.024	4
1	100b	Fhosc/8	0.128	0.5	0.512	2
1	101b	Fhosc/4	0.064	0.25	0.256	1
1	110b	Fhosc/2	0.032	0.125	0.128	0.5
1	111b	Fhosc/1	0.016	0.0625	0.064	0.25

3.3.3.3 TC1M MODE REGISTER

TC1M is TC1 timer mode control register to configure TC1 operating mode including TC1 pre-scaler, clock source, PWM function…These configurations must be setup completely before enabling TC1 timer.

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1ENB	TC1rate2	TC1rate1	TC1rate0	TC1CKS	ALOAD1	TC1OUT	PWM1OUT

Bit<7>:**TC1ENB**-TC1 counter control bit.
0:Disable
1:Enable

Bit<6:4>:**TC1rate<2:0>**:TC1 internal clock select bits.

TC1rate<2:0>	TC1X8=0	TC1X8=1
000	Fcpu/256	Fosc/128
001	Fcpu/128	Fosc/64
010	Fcpu/64	Fosc/32
011	Fcpu/32	Fosc/16
100	Fcpu/16	Fosc/8
101	Fcpu/8	Fosc/4
110	Fcpu/4	Fosc/2
111	Fcpu/2	Fosc/1

Bit<3>:**TC1CKS**- TC1 clock source select bit.
0:Internal clock (Fcpu and Fhosc controlled by TC1X8 bit).
1:External input pin (P0.1/INT1) and enable event counter function, TC1rate[2:0] bits are useless.

Bit<2>:**ALOAD1**- Auto-reload control bit. Only valid when PWM1OUT = 0.
0: Disable
1: Enable

Bit<1>:**TC1OUT**- TC1 time out toggle signal output control bit.
(Only valid when PWM1OUT = 0.)
0:Disable, P5.3 is GPIO mode.
1:Enable, P5.3 outputs TC1/2 buzzer signal.

Bit<0>:**PWM1OUT**- PWM output control bit.
0:Disable, P5.3 is GPIO mode.
1:Enable, P5.3 outputs PWM signal. PWM duty controlled by TC1OUT, ALOAD1 bits.

Note: When TC1CKS=1, TC1 became an external event counter and TC1RATE is useless. No more P0.1 interrupt request will be raised. (P0.1IRQ will be always 0).

3.3.3.4 TC1X8/TC1GN FLAG

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	--	TC1X8	TC0X8	TC0GN	--

Bit<3>:**TC1X8**- TC1 internal clock source control bit.
0 = TC1 internal clock source is Fcpu. TC1RATE is from Fcpu/2～Fcpu/256.
1 = TC1 internal clock source is Fhosc. TC1RATE is from Fhosc/1～Fhosc/128.

Note: Under TC1 event counter mode (TC1CKS=1), TC1X8 bit and TC1RATE are useless.

3.3.3.5 TC1C COUNTING REGISTER

TC1C is TC1 8-bit counter. When TC1C overflow occurs, the TC1IRQ flag is set as "1" and cleared by program. The TC1C decides TC1 interval time through below equation to calculate a correct value. It is necessary to write the correct value to TC1C register and TC1R register, and then enable TC1 timer. After TC1 overflow occurs, the TC1C register is loaded a correct value from TC1R register automatically, not program.

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1C7	TC1C6	TC1C5	TC1C4	TC1C3	TC1C2	TC1C1	TC1C0

The equation of TC1C initial value is as following:

TC1C initial value = N - (TC1 interrupt interval time * input clock)

N is TC1 overflow boundary number. The setting of parameters in each mode is shown in the following table:

TC1CKS	TC1X8	PWM1	ALOAD1	TC1OUT	N	TC1C Value
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

3.3.3.6 TC1R AUTO-LOAD REGISTER

TC1 timer builds in auto-reload function, and TC1R register stores reload data. When TC1C overflow occurs, TC1C register is loaded data from TC1R register automatically. Under TC1 timer counting status, to modify TC1 interval time is to modify TC1R register, not TC1C register. New TC1C data of TC1 interval time will be updated after TC1 timer overflow occurrence, TC1R loads new value to TC1C register. But at the first time to setup TC1M, TC1C and TC1R must be set the same value before enabling TC1 timer. TC1 is double buffer design. If new TC1R value is set by program, the new value is stored in 1st buffer. Until TC1 overflow occurs, the new value moves to real TC1R buffer. This way can avoid any transitional condition to effect the correctness of TC1 interval time and PWM output signal.

Note: Under PWM mode, auto-load is enabled automatically. The ALOAD1 bit is selecting overflow boundary.

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC1R7	TC1R6	TC1R5	TC1R4	TC1R3	TC1R2	TC1R1	TC1R0

The equation of TC1R initial value is as following:
TC1R initial value = N - (TC1 interrupt interval time * input clock)
N is TC1 overflow boundary number. TC1 overflow time and valid value as follow table:

TC1CKS	TC1X8	PWM1	ALOAD1	TC1OUT	N	TC1R Value
0	0 (Fcpu/2~ Fcpu/256)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
	1 (Fosc/1~ Fosc/128)	0	×	×	256	00H~0FFH
		1	0	0	256	00H~0FFH
		1	0	1	64	00H~3FH
		1	1	0	32	00H~1FH
		1	1	1	16	00H~0FH
1	-	-	-	-	256	00H~0FFH

Example: To set 10ms interval time for TC1 interrupt. TC1 clock source is Fcpu (TC1KS=0, TC1X8=0) and no PWM output (PWM1=0). High clock is external 4MHz. Fcpu=Fhosc/4. Select TC1RATE=010 (Fcpu/64).

TC1R = N - (TC1 interrupt interval time * input clock)
= 256 - (10ms * 4MHz / 4 / 64)
= 256 - (0.01* 4000000/ 4 / 64)
= 100 = 64H

3.3.3.7 TC1 EVENT COUNTER FUNCTION

TC1 event counter is set the TC1 clock source from external input pin (P0.1). When TC1CKS=1, TC1 clock source is switch to external input pin (P0.1). TC1 event counter trigger direction is falling edge. When one falling edge occurs, TC1C will up one count. When TC1C counts from 0xFF to 0x00, TC1 triggers overflow event. The external event counter input pin's wake-up function of GPIO mode is disabled when TC1 event counter function enabled to avoid event counter signal trigger system wake-up and not keep in power saving mode. The external event counter input pin's external interrupt function is also disabled when TC1 event counter function enabled, and the P01IRQ bit keeps "0" status. The event counter usually is used to measure external continuous signal rate, e.g. continuous pulse, R/C type oscillating signal. These signal phase don't synchronize with MCU's main clock. Use TC1 event to measure it and calculate the signal rate in program for different applications.

3.3.3.8 TC1 CLOCK FREQUENCY OUTPUT

Buzzer output (TC1OUT) is from TC1 timer/counter frequency output function. By setting the TC1 clock frequency, the clock signal is output to P5.3 and the P5.3 general purpose I/O function is auto-disable. The TC1OUT frequency is divided by 2 from TC1 interval time. TC1OUT frequency is 1/2 TC1 frequency. The TC1 clock has many combinations and easily to make difference frequency.

When buzzer outputs, TC1IRQ still actives as TC1 overflows, and TC1 interrupt function actives as TC1IEN = 1. But strongly recommend be careful to use buzzer and TC1 timer together, and make sure both functions work well. The buzzer output pin

is shared with GPIO and switch to output buzzer signal as TC1OUT=1 automatically. If TC1OUT bit is cleared to disable buzzer signal, the output pin exchanges to last GPIO mode automatically.

Example: Setup TC1OUT output from TC1 to TC1OUT (P5.3). The external high-speed clock Fhosc is 4MHz. the instruction cycle Fcpu is Fhosc/4. The TC1OUT frequency is 0.5KHz. Because the TC1OUT signal is divided by 2, set the TC1 clock to 1KHz. The TC1 clock source is from external oscillator clock. TC1 rate is Fcpu/8.

TC1RATE2～TC1RATE0=101, TC1C=TC1R=131。

```
MOV     A,#01010000B
BOMOV   TC1M,A          ; Set the TC1 rate to Fcpu/8
MOV     A,#131          ; Set the auto-reload reference value
BOMOV   TC1C,A
BOMOV   TC1R,A
BOBSET FTC1OUT          ; Enable TC1 output to P5.3 and disable P5.3 I/O function
BOBSET FAL1AD0          ; Enable TC1 auto-reload function
BOBSET FTC1ENB          ; Enable TC1 timer
```

Note: Buzzer output is enable, and “PWM1OUT” must be “0”.

3.3.3.9 PULSE WIDTH MODULATION (PWM)

The PWM is duty/cycle programmable design to offer various PWM signals. When TC1 timer enables and PWM1OUT bit sets as “1” (enable PWM outputs), the PWM output pin outputs PWM signal. One cycle of PWM signal is high pulse first, and then low pulse outputs. TC1RATE, ALOAD1, TC1OUT bits control the cycle of PWM, and TC1R decides the duty (high pulse width length) of PWM. TC1C initial value must be set to zero when TC1 timer enables. When TC1C count is equal to TC1R, the PWM high pulse finishes and exchanges to low level. When TC1 overflows, one complete PWM cycle finishes. The PWM exchanges to high level for next cycle. If modify the PWM cycle by program as PWM outputting, the new cycle occurs at next cycle when TC1C loaded from TC1R.

The PWM builds in four programmable resolution (1/256, 4/64, 4/32, 4/16) controlled by ALOAD1 and TC1OUT bits as PWM1OUT = 1.

PWM1	ALOAD1	TC1OUT	PWM Resolution	TC1R valid value	TC1R value binary type
1	0	0	256	00H~0FFH	00000000B~11111111B
1	0	1	64	00H~3FH	xx000000B~xx111111B
1	1	0	32	00H~1FH	xxx00000B~xxx11111B
1	1	1	16	00H~0FH	xxxx0000B~xxxx1111B

TC1R controls the high pulse width of PWM for PWM’s duty. When TC1C = TC1R, PWM output exchanges to low status. When PWM outputs, TC1IRQ still actives as TC1 overflows, and TC1 interrupt function actives as TC1IEN = 1. TC1 interrupt interval time is equal to PWM’s cycle in PWM mode. That means TC1 interrupt period has four resolution following ALOAL1 and TC1OUT values. But strongly recommend be careful to use PWM and TC1 timer together, and make sure both functions work well.

The PWM output pin is shared with GPIO and switch to output PWM signal as PWM1OUT=1 automatically. If PWM1OUT bit is cleared to disable PWM, the output pin exchanges to last GPIO mode automatically. It easily to implement carry signal on/off operation, not to control TC1ENB bit.

3.4 ANALOG TO DIGITAL CONVERTER

3.4.1 OVERVIEW

The analog to digital converter (ADC) is SAR structure with 6-input sources and 4096-step resolution to transfer analog signal into 12-bits digital data. Use CHS[2:0] bits to select analog signal input pin (AIN pin), internal 1/4*Vdd voltage source, and GCHS bit enables global ADC channel, the analog signal inputs to the SAR ADC. The ADC resolution is 12-bit resolutions. The ADC converting rate can be selected by ADCKS[1:0] bits to decide ADC converting time. The ADC reference high voltage includes two sources. One is internal multi-level source including Vdd, 4V, 3V, 2V (EVHENB=0), and the other one is external reference voltage input pin from P4.0 pin (EVHENB=1). The ADC also builds in P4CON register to set pure analog input pin. It is necessary to set P4 as input mode with pull-up resistor by program. After setup ADENB and ADS bits, the ADC starts to convert analog signal to digital data. When the conversion is complete, the ADC circuit will set EOC and ADCIRQ bits to "1" and the digital data outputs in ADB and ADR registers. If the ADCIEN = 1, the ADC interrupt request occurs and executes interrupt service routine when ADCIRQ = 1 after ADC converting.

Note:

(1) Set ADC input pin I/O direction as input mode without pull-up resistor.

(2) Disable ADC (set ADENB = "0") before enter sleep mode to save power consumption.

(3) Set related bit of P4CON register to avoid extra power consumption in power down mode.

(4) Delay 100uS after enable ADC (set ADENB = "1") to wait ADC circuit ready for conversion.

(5) When VDD is used as the reference voltage for AD on JZ8P2612, the maximum value cannot reach 4095, but can reach about 4040. This value does not affect the normal use of the chip, but is only used when judging NTC short circuit/open circuit. And the higher the temperature, the smaller the value. If the internal reference voltage of 2V, 3V, 4V is selected, it will not affect.

3.4.2 ADC MODE REGISTER

ADM is ADC mode control register to configure ADC configurations including ADC start, ADC channel selection, ADC high reference voltage source and ADC processing indicator. These configurations must be setup completely before starting ADC converting.

0B1H (ADM CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADENB	ADS	EOC	GCHS	--	CHS2	CHS1	CHS0

Bit<7>:**ADENB**-ADC control bit. In sleep mode, disable ADC to reduce power consumption.

0: Disable

1: Enable

Bit<6>:**ADS**-ADCADC start control bit. ADS bit is cleared after ADC processing automatically.

0: ADC converting stops.

1: Start to execute ADC converting.

Bit<5>:**EOC**-ADC status bit.

0: ADC progressing.

1: End of converting and reset ADS bit.

Bit<4>:**GCHS**-ADC global channel select bit.

0: Disable

1: Enable

Bit<2:0>: **CHS<2:0>**-ADC input channel select bit.

CHS<2:0>	Channel	Port
000	AIN0	P40
001	AIN1	P41
010	AIN2	P42
011	AIN3	P43
100	AIN4	P44
101	AIN5	Internal 1/4VDD channel

The AIN5 is internal 1/4*VDD input channel. There is no any input pin from outside. AIN5 can be a good battery detector for battery system. To select appropriate internal VREFH level and compare value, a high performance and cheaper low battery detector is built in the system.

0B2H (ADC DATA BUFFER REGISTERS)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4

Bit<7:0>:**ADB<11:4>**-ADC data is stored in ADB (read only)

DC data buffer is 12-bit length to store ADC converter result. The high byte is ADB register, and the low-nibble is ADR[3:0] bits. The ADB register is only 8-bit register including bit 4～bit11 ADC data. To combine ADB register and the low-nibble of ADR will get full 12-bit ADC data buffer. The ADC data buffer is a read-only register and the initial status is unknown after system reset.

0B3H (ADC PRECISION SELECTION REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	ADCKS1	-	ADCKS0	ADB3	ADB2	ADB1	ADB0

Bit<6,4>:**ADCKS<1:0>**-ADC clock source selector bit.

ADCKS1	ADCKS0	ADC Clock Source
0	0	Fcpu/16
0	1	Fcpu/8
1	0	Fcpu
1	1	Fcpu/2

Bit<3:0>:**ADB<3:0>**- ADC data is stored in ADR

3.4.3 ADC PIN CONFIGURATION

ADC input channels are shared with Port4. ADC channel selection is through CHS[2:0] bit. CHS[2:0] value points to the ADC input channel directly. CHS[2:0]=000 selects AIN0. CHS[2:0]=001 selects AIN1…… Only one pin of port4 can be configured as ADC

input in the same time. The pins of Port4 configured as ADC input channel must be set input mode, disable internal pull-up and enable P4CON first by program. After selecting ADC input channel through CHS[2:0], set GCHS bit as "1" to enable ADC channel function.

Note:

The GPIO mode of ADC input channels must be set as input mode.

The internal pull-up resistor of ADC input channels must be disabled.

P4CON bits of ADC input channel must be set.

The P4.0/AIN0 can be ADC external high reference voltage input pin when AVREFH=1. In the condition, P4.0 GPIO mode must be set as input mode and disable internal pull-up resistor.

Note:

The GPIO mode of ADC external high reference voltage input pin must be set as input mode.

The internal pull-up resistor of ADC external high reference voltage input pin must be disabled.

ADC input pins are shared with digital I/O pins. Connect an analog signal to COMS digital input pin, especially, the analog signal level is about 1/2 VDD will cause extra current leakage. In the sleep mode, the above leakage current will be a big problem. Unfortunately, if users connect more than one analog input signal to port4 will encounter above current leakage situation. P4CON is Port4 configuration register. Write "1" into P4CON [4:0] will configure related port 4 pin as pure analog input pin to avoid current leakage.

0AEH (P4CON PORT CONFIGURATION CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P4CON4	P4CON3	P4CON2	P4CON1	P4CON0

Bit<4:0>:P4CON<4:0>-P4 ADC configuration control bits.

0:P4 can be an analog input (ADC input) or digital I/O pins.

1:P4 is pure analog input, can't be a digital I/O pin.

3.4.4 ADC REFERENCE VOLTAGE REGISTER

ADC builds in five high reference voltage source controlled through VREFH register. There are one external voltage source and four internal voltage source (VDD, 4V, 3V, 2V). When EVHENB bit is "1", ADC reference voltage is external voltage source from P4.0. It is necessary to input a voltage to be ADC high reference voltage and not below 2V. If EVHENB bit is "0", ADC reference voltage is from internal voltage source selected by VHS[1:0] bits. If VHS[1:0] is "11", ADC reference voltage is VDD. If VHS[1:0] is "10", ADC reference voltage is 4V. If VHS[1:0] is "01", ADC reference voltage is 3V. If VHS[1:0] is "00", ADC reference voltage is 2V. The limitation of internal reference voltage application is VDD can"t below each of internal voltage level, or the level is equal to VDD.

0AFH (ADC REFERENCE VOLTAGE REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EVHENB	-	-	-	-	-	VHS1	VHS0

Bit<7>:**EVHENB**- ADC internal reference high voltage control bit.

0: Enable ADC internal VREFH function, P4.0/AIN0/VREFH pin is P4.0/AIN0 pin.

1: Disable ADC internal VREFH function, P4.0/AIN0/VREFH pin is external VREFH input pin.

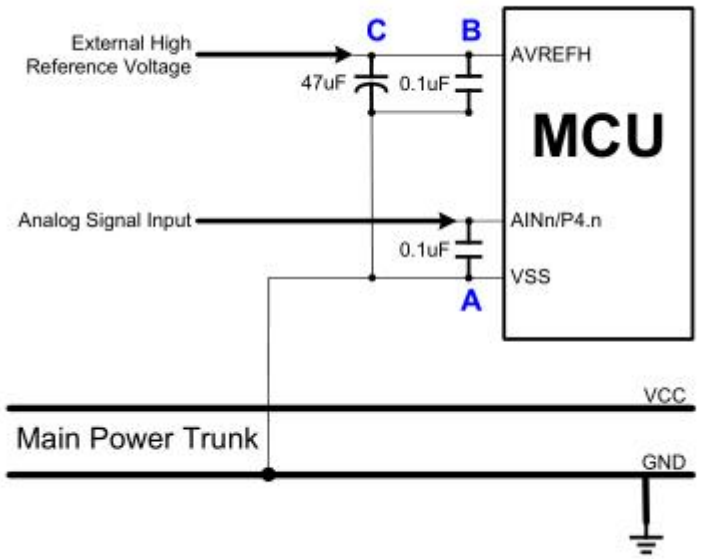
Bit<1:0>:VHS<1:0>- ADC internal reference high voltage select bits.

VHS1	VHS0	Internal VREFH Voltage
0	0	2.0V
0	1	3.0V
1	0	4.0V
1	1	VDD

3.4.5 ADC APPLICATION CIRCUIT

The analog signal is inputted to ADC input pin "AINn/P4.n". The ADC input signal must be through a 0.1uF capacitor "A". The 0.1uF capacitor is set between ADC input pin and VSS pin, and must be on the side of the ADC input pin as possible. Don't connect the capacitor's ground pin to ground plain directly, and must be through VSS pin. The capacitor can reduce the power noise effective coupled with the analog signal.

If the ADC high reference voltage is from external voltage source, the external high reference is connected to AVREFH pin (P4.0). The external high reference source must be through a 47uF "C" capacitor first, and then 0.1uF capacitor "B". These capacitors are set between AVREFH pin and VSS pin, and must be on the side of the AVREFH pin as possible. Don't connect the capacitor"s ground pin to ground plain directly, and must be through VSS pin.



3.4.6 ADC OPERATION EXAMLPE

1. Reset ADC

```
CLR     ADM             ; Clear ADM register.
```

2. Set ADC clock rate and ADC resolution.

```
    MOV     A, #0nmn0000b       ;nn :ADCKS[1:0] for ADC clock rate.
    BOMOV   ADR, A
```

3. Set ADC high reference voltage source

```
    BOBSET FEVHENB          ; External reference voltage.
```

or

```
    MOV     A, #000000nnb     ; Internal Vdd.
                              ; “nn” select internal reference voltage level.
                              ; 11 = VDD, 10 = 4V, 01 = 3V, 00 = 2V。
```

4. Set ADC input channel configuration.

```
    MOV     A, #value1        ; Set P4CON for ADC input channel.
    BOMOV   P4CON, A
    MOV     A, #value2        ; Set ADC input channel as input mode.
    BOMOV   P4M, A
    MOV     A, #value3        ;Disable ADC input channel"s internal pull-up resistor.
    BOMOV   P4UR, A
```

5. Enable ADC

```
    BOBSET  FADCENB
```

6. Execute ADC 100us warm-up time delay loop

```
    CALL    100usDLY    ; 100us delay loop.
```

7. Select ADC input channel

```
    MOV     A, #value         ; Set CHS[2:0] for ADC input channel selection.
    OR      ADM, A
```

8. Enable ADC input channel

```
    BOBSET  FGCHS
```

9. Enable ADC interrupt function

```
    BOBCLR  FADCIRQ   ; Clear ADC interrupt flag.
    BOBSET  FADCIEN   ; Enable ADC interrupt function.
```

10. Start to execute ADC converting

```
    BOBSET  FADS
```

Note:

- 1、When ADENB is enabled, the system must be delay 100us to be the ADC warm-up time by program, and then set ADS to do ADC converting. The 100us delay time is necessary after ADENB setting (not ADS setting), or the ADC converting result would be error. Normally, the ADENB is set one time when the system under normal run condition, and do the delay time only one time.
- 2、In sleep mode situation like power down mode and green mode, and not using ADC function, to disable ADC by program is necessary to reduce power consumption.

3.5 LOW VOLTAGE DETECTOR

The LVD (low voltage detector) is built-in JZ8P2612 8-bit MCU to be brown out reset protection. When the VDD drops and is below LVD detect voltage, the LVD would be triggered, and the system is reset. The LVD detect level is different by each MCU. The LVD voltage level is a point of voltage and not easy to cover all dead-band range. Using LVD to improve brown out reset is depend on application requirement and environment. If the power variation is very deep, violent and trigger the LVD, the LVD can be the protection. If the power variation can touch the LVD detect level and make system work error, the LVD can"t be the protection and need to other reset methods.

The LVD is two levels design (2.4V/3.6V) and controlled by LVD code option. The 2.4V LVD includes LVD reset function and flag function to indicate VDD status function. The 3.6V includes flag function to indicate VDD status. LVD flag function can be an easy low battery detector. LVD24, LVD36 flags indicate VDD voltage level. For low battery detect application, only checking LVD24, LVD36 status to be battery status. This is a cheap and easy solution.

3.5.1 LVD VOLTAGE DETECTION REGISTER

086H (PFLAG PROGRAM FLAG)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NTO	NPD	LVD36	LVD24	-	C	DC	Z

Bit<5>: **LVD36**- LVD 3.6V operating flag and only support LVD code option is LVD_H.
0:Inactive (VDD > 3.6V).
1: Active (VDD ≦ 3.6V).

Bit<4>: **LVD36**- LVD 2.4V operating flag and only support LVD code option is LVD_H.
0:Inactive (VDD > 2.4V).
1: Active (VDD ≦ 2.4V).

LVD	LVD Compiler options		
	LVD_L	LVD_M	LVD_H
2.0V reset	Enable	Enable	Enable
2.4V flag	-	Enable	-
2.4V reset	-	-	Enable
3.6V flag	-	-	Enable

LVD_L: If VDD is less than 2.0V, the system is reset;
The flags of LVD24 and LVD36 are meaningless.

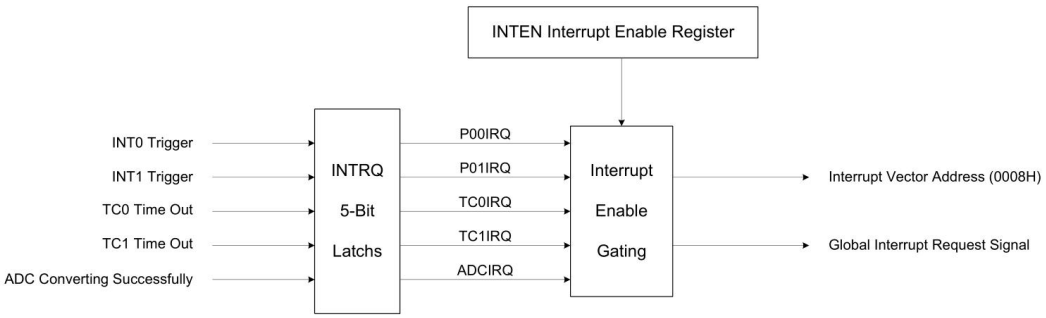
LVD_M: If VDD is less than 2.0V, the system is reset;
LVD24:If VDD > 2.4V, LVD24 =0; Otherwise, VDD <= 2.4V, LVD24=1;
LVD36: meaningless.

LVD_H: If VDD is less than 2.4V, the system is reset;
LVD36:If VDD > 3.6V, LVD36=0;Otherwise, VDD <= 3.6V, LVD36=1;

Note:
(1) When LVD reset ends, LVD24 and LVD36 will both be cleared;
(2) LVD 2.4V and LVD 3.6V detection level values are only used as design reference and cannot be used for accurate detection of chip operating voltage .

3.6 INTERRUPT

JZ8P2612 provides five interrupt sources, including three internal interrupt (TC0/TC1/ADC) and two external interrupt (INT0/INT1). The external interrupt can wake-up the chip while the system is switched from sleep mode to high-speed normal mode, and interrupt request is latched until return to normal mode. Once interrupt service is executed, the GIE bit in STKP register will clear to "0" for stopping other interrupt request. On the contrast, when interrupt service exits, the GIE bit will set to "1" to accept the next interrupts' request. All of the interrupt request signals are stored in INTRQ register.



3.6.1 INTERRUPT ENABLE REGISTER

INTEN is the interrupt request control register including three internal interrupts, two external interrupts enable control bits. One of the register to be set "1" is to enable the interrupt request function. Once of the interrupt occur, the stack is incremented and program jump to ORG 8 to execute interrupt service routines. The program exits the interrupt service routine when the returning interrupt service routine instruction (RETI) is executed.

0C9H (INTEN INTERRUPT ENABLE REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCIEN	TC1IEN	TC0IEN	-	-	-	P01IEN	P00IEN

Bit<7>:**ADCIEN**- ADC interrupt control bit.
Bit<6>:**TC1IEN**- TC1 timer interrupt control bit.
Bit<5>:**TC0IEN**- TC0 timer interrupt control bit.
Bit<1>:**P01IEN**- External P0.1 interrupt (INT1) control bit.
Bit<0>:**P00IEN**- External P0.0 interrupt (INT0) control bit.
0: Disable
1: Enable

3.6.2 INTERRUPT REQUEST REGISTER

INTRQ is the interrupt request flag register. The register includes all interrupt request indication flags. Each one of the interrupt requests occurs, the bit of the INTRQ register would be set "1". The INTRQ value needs to be clear by programming after detecting the flag. In the interrupt vector of program, users know the any interrupt requests occurring by the register and do the routine corresponding of the interrupt request.

0C8H (INTRQ INTERRUPT REQUEST REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCIRQ	TC1IRQ	TC0IRQ	-	-	-	P01IRQ	P00IRQ

Bit<7>:**ADCIRQ**-ADC interrupt request flag.
Bit<6>:**TC1IEN**-TC1 timer interrupt request flag.
Bit<5>:**TC0IEN**-TC0 timer interrupt request flag.
Bit<1>:**P01IEN**-External P0.1 interrupt (INT1) request flag.
Bit<0>:**P00IEN**-External P0.0 interrupt (INT0) request flag.
0: None interrupt request.
1: Interrupt request.

3.6.3 GIE GLOBAL INTERRUPT OPERATION

GIE is the global interrupt control bit. All interrupts start work after the GIE = 1 It is necessary for interrupt service request. One of the interrupt requests occurs, and the program counter (PC) points to the interrupt vector (ORG 8) and the stack add 1 level.

0DFH (STKP CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GIE	-	-	-	-	STKPB2	STKPB1	STKPB0

Bit<7>:**GIE**- Global interrupt control bit.
0: Disable
1: Enable

Note: The GIE bit must enable during all interrupt operation.

3.6.4 INT0 (P0.0) EXTERNAL INTERRUPT OPERATION

INT0 is external interrupt trigger source and builds in edge trigger configuration function. When the external edge trigger occurs, the external interrupt request flag will be set to "1" no matter the external interrupt control bit enabled or disable. When external interrupt control bit is enabled and external interrupt edge trigger is occurring, the program counter will jump to the interrupt vector (ORG 8) and execute interrupt service routine.

Note: INT0 interrupt request is locked by P0.0 wake trigger function, P0.0 interrupt trigger edge is controlled by PEDGE.

0BFH (PEDGE CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
-	-	-	P00G1	P00G0	-	-	-

Bit<4:3>:**P00G<1:0>**-P00 edge trigger select bits.
00: reserved.
01: rising edge.
10: falling edge.
11: rising/falling bi-direction.

3.6.5 INT1 (P0.1) INTERRUPT OPERATION

When the INT1 trigger occurs, the P01IRQ will be set to "1" no matter the P01IEN is enable or disable. If the P01IEN = 1 and the trigger event P01IRQ is also set to be "1". As the result, the system will execute the interrupt vector (ORG 8). If the P01IEN = 0 and the trigger event P01IRQ is still set to be "1". Moreover, the system won't execute interrupt vector even when the P01IRQ is set to be "1". Users need to be cautious with the operation under multi-interrupt situation.

If the interrupt trigger direction is identical with wake-up trigger direction, the INT1 interrupt request flag (INT1IRQ) is latched while system wake-up from power down mode or green mode by P0.1 wake-up trigger. System inserts to interrupt vector (ORG 8) after wake-up immediately.

Note: INT1 interrupt request is locked by P0.1 wake trigger function, P0.1 interrupt trigger edge is controlled by PEDGE.

3.6.6 TC0 INTERRUPT

When the TC0C counter overflows, the TC0IRQ will be set to "1" no matter the TC0IEN is enable or disable. If the TC0IEN and the trigger event TC0IRQ is set to be "1". As the result, the system will execute the interrupt vector. If the TC0IEN = 0, the trigger event TC0IRQ is still set to be "1". Moreover, the system won't execute interrupt vector even when the TC0IEN is set to be "1". Users need to be cautious with the operation under multi-interrupt situation.

TC0 interrupt request setup:

```
BOBCLR FTC0IEN    ; Disable TC0 interrupt service
BOBCLR FTC0ENB
MOV    A, #20H
BOMOV  TC0M, A    ; Set TC0 clock = Fcpu / 64
MOV    A, # 64H   ; Set TC0C initial value = 64H
BOMOV  TC0C, A    ; Set TC0 interval = 10 ms
BOBCLR FTC0IRQ    ; Clear TC0 interrupt request flag
BOBSET FTC0IEN    ; Enable TC0 interrupt service
BOBSET FTC0ENB
BOBSET FGIE       ; Enable GIE
```

3.6.7 TC1 INTERRUPT

When the TC1C counter overflows, the TC1IRQ will be set to "1" no matter the TC1IEN is enable or disable. If the TC1IEN and the trigger event TC1IRQ is set to be "1". As the result, the system will execute the interrupt vector. If the TC1IEN = 0, the trigger event TC1IRQ is still set to be "1". Moreover, the system won't execute interrupt vector even when the TC1IEN is set to be "1". Users need to be cautious with the operation under multi-interrupt situation.

TC1 interrupt request setup:

```
BOBCLR FTC1IEN    ; Disable TC1 interrupt service
BOBCLR FTC1ENB
MOV    A, #20H
BOMOV  TC1M, A    ; Set TC1 clock = Fcpu / 64
```

```
MOV     A, # 64H    ; Set TC1C initial value = 64H
BOMOV   TC1C, A     ; Set TC1 interval = 10 ms
BOBCLR  FTC1IRQ     ; Clear TC1 interrupt request flag
BOBSET  FTC1IEN     ; Enable TC1 interrupt service
BOBSET  FTC1ENB     ; Enable TC1 timer
BOBSET  FGIE        ; Enable GIE
```

3.6.8 ADC INTERRUPT

When the ADC converting successfully, the ADCIRQ will be set to "1" no matter the ADCIEN is enable or disable. If the ADCIEN and the trigger event ADCIRQ is set to be "1". As the result, the system will execute the interrupt vector. If the ADCIEN = 0, the trigger event ADCIRQ is still set to be "1". Moreover, the system won't execute interrupt vector even when the ADCIEN is set to be "1". Users need to be cautious with the operation under multi-interrupt situation.

ADC interrupt request setup:

```
BOBCLR  FADCIEN     ; Disable ADC interrupt service
MOV A,  #10110000B
BOMOV   ADM, A      ; Enable P4.0 ADC input and ADC function
MOV A,  #00000000B  ; Set ADC converting rate = Fcpu/16
BOMOV   ADR, A
BOBCLR  FADCIRQ     ; Clear ADC interrupt request flag
BOBSET  FADCIEN     ; Enable ADC interrupt service
BOBSET  FGIE        ; Enable GIE
BOBSET  FADS        ; Start ADC transformation
```

3.6.9 MULTI-INTERRUPT OPERATION

Under certain condition, the software designer uses more than one interrupt requests. Processing multi-interrupt request requires setting the priority of the interrupt requests. The IRQ flags of interrupts are controlled by the interrupt event. Nevertheless, the IRQ flag "1" doesn't mean the system will execute the interrupt vector. In addition, which means the IRQ flags can be set "1" by the events without enable the interrupt. Once the event occurs, the IRQ will be logic "1". The IRQ and its trigger event relationship is as the below table:

Interrupt	Trigger Event
P00IRQ	P0.0 trigger controlled by PEDGE
P01IRQ	P0.1 falling edge trigger
TC0IRQ	TC0C overflow
TC1IRQ	TC1C overflow
ADCIRQ	ADC converting successfully

For multi-interrupt conditions, two things need to be taking care of. One is to set the priority for these interrupt requests. Two is using IEN and IRQ flags to decide which interrupt to be executed. Users have to check interrupt control bit and interrupt request flag in interrupt routine.

Example: Check the interrupt request under multi-interrupt operation

```
    ORG     8H  ;
```

```
        JMP     INT_SERVICE
INT_SERVICE:
        …                   ; Push routine to save ACC and PFLAG to buffers
INTP00CHK:                  ; Check INT0 interrupt request
        B0BTS1 FP00IEN      ; Check P00IEN
        JMP     INTP01CHK   ; Jump check to next interrupt
        B0BTS0 FP00IRQ      ; Check P00IRQ
        JMP     INTP00      ; Jump to INT0 interrupt service routine
INTP01CHK:                  ; Check INT0 interrupt request
        B0BTS1 FP01IEN      ; Check P01IEN
        JMP     INTTC0CHK   ; Jump check to next interrupt
        B0BTS0 FP01IRQ      ; Check P01IRQ
        JMP     INTP01      ; Jump to INT1 interrupt service routine
INTTC0CHK:                  ; Check TC0 interrupt request
        B0BTS1 FTC0IEN      ; Check TC0IEN
        JMP     INTTC1CHK   ; Jump check to next interrupt
        B0BTS0 FTC0IRQ      ; Check TC0IRQ
        JMP     INTTC0      ; Jump to TC0 interrupt service routine
INTTC1CHK:                  ; Check TC1 interrupt request。
        B0BTS1 FTC1IEN      ; Check TC1IEN
        JMP     INTADCHK    ; Jump check to next interrupt
        B0BTS0 FTC1IRQ      ; Check TC1IRQ
        JMP     INTTC1      ; Jump to TC1 interrupt service routine
INTADCHK:                   ; Check ADC interrupt request
        B0BTS1 FADCIEN      ; Check ADCIEN
        JMP     INT_EXIT    ;
        B0BTS0 FADCIRQ      ; Check ADCIRQ
        JMP     INTADC      ;Jump to ADC interrupt service routine
INT_EXIT:
        …                   ; Pop routine to load ACC and PFLAG from buffers
        RETI                ; Exit interrupt vector
```

3.7 RESET

3.7.1 OVERVIEW

JZ8P2612 would be reset in four conditions as following:

1、Power on reset;

2、Watchdog reset;

3、Brown out reset;

4、External reset (only supports external reset pin enable situation);

When any reset condition occurs, all system registers keep initial status, program stops and program counter is cleared. After reset status released, the system boots up and program starts to execute from ORG 0000H. The NTO, NPD flags indicate system reset status. The system can depend on NTO, NPD status and go to different paths by program.

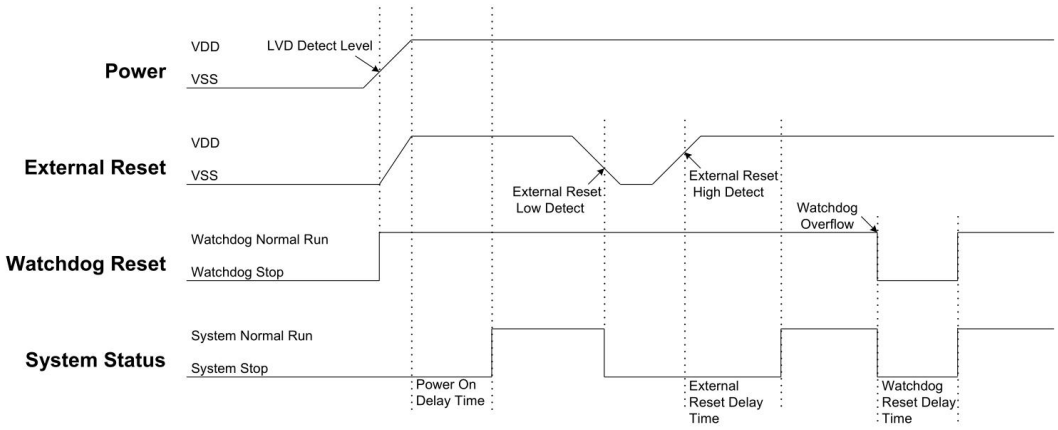
0B6H (PFLAG CONTROL REGISTER)

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
NTO	NPD	LVD36	LVD24		C	DC	Z

Bit<7:6>:NPO/NPD- Reset status flag.

NTO	NPD	Condition	Description
0	0	Watchdog reset	Watchdog timer overflow
0	1	Reserved	-
1	0	Power on reset and LVD reset	Power voltage is lower than LVD detecting level
1	1	External reset	External reset pin detect low level status

Finishing any reset sequence needs some time. The system provides complete procedures to make the power on reset successful. For different oscillator types, the reset time is different. That causes the VDD rise rate and start-up time of different oscillator is not fixed. RC type oscillator’s start-up time is very short, but the crystal type is longer. Under client terminal application, users have to take care the power on reset time for the master terminal requirement. The reset timing diagram is as following.



3.7.2 POWER ON RESET

The power on reset depend no LVR operation for most power-up situations. The power supplying to system is a rising curve and needs some time to achieve the normal voltage. Power on reset sequence is as following.

Power-up: System detects the power voltage up and waits for power stable.

External reset (only external reset pin enable): System checks external reset pin status. If external reset pin is not high level, the system keeps reset status and waits external reset pin released.

System initialization: All system registers is set as initial conditions and system is ready.

Oscillator warm up: Oscillator operation is successfully and supply to system clock.

Program executing: Power on sequence is finished and program executes from ORG 000H.

3.7.3 WATCHDOG RESET

Watchdog reset is a system protection. In normal condition, system works well and clears watchdog timer by program. Under error condition, system is in unknown situation and watchdog can't be clear by program before watchdog timer overflow. Watchdog timer overflow occurs and the system is reset. After watchdog reset, the system restarts and returns normal mode. Watchdog reset sequence is as following.

Watchdog timer status: System checks watchdog timer overflow status. If watchdog timer overflow occurs, the system is reset.

System initialization: All system registers is set as initial conditions and system is ready.

Oscillator warm up: Oscillator operation is successfully and supply to system clock.

Program executing: Power on sequence is finished and program executes from ORG 000H.

Watchdog timer application note is as following:

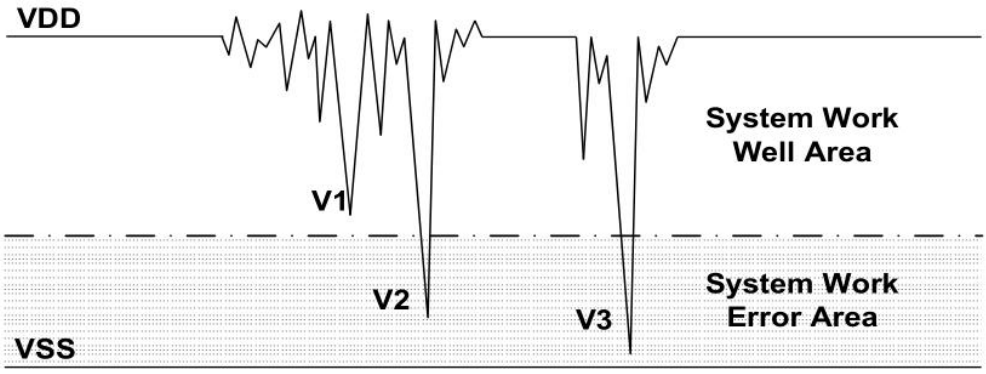
1、Before clearing watchdog timer, check I/O status and check RAM contents can improve system error.

2、Don't clear watchdog timer in interrupt vector and interrupt service routine. That can improve main routine fail.

3、Clearing watchdog timer program is only at one part of the program. This way is the best structure to enhance the watchdog timer function.

3.7.4 BROWN OUT RESET

The brown out reset is a power dropping condition. The power drops from normal voltage to low voltage by external factors (e.g. EFT interference or external loading changed). The brown out reset would make the system not work well or executing program error.



The power dropping might through the voltage range that's the system dead-band. The dead-band means the power range can't offer the system minimum operation power requirement. The above diagram is a typical brown out reset diagram. There is a serious noise under the VDD, and VDD voltage drops very deep. There is a dotted line to separate the system working area. The above area is the system work well area. The below area is the system work error area called dead-band. V1 doesn't touch the below area and not effect the system operation. But the V2 and V3 is under the below area and may induce the system error occurrence. Let system under dead-band includes some conditions.

DC application:

The power source of DC application is usually using battery. When low battery condition and MCU drive any loading, the power drops and keeps in dead-band. Under the situation, the power won't drop deeper and not touch the system reset voltage. That makes the system under dead-band.

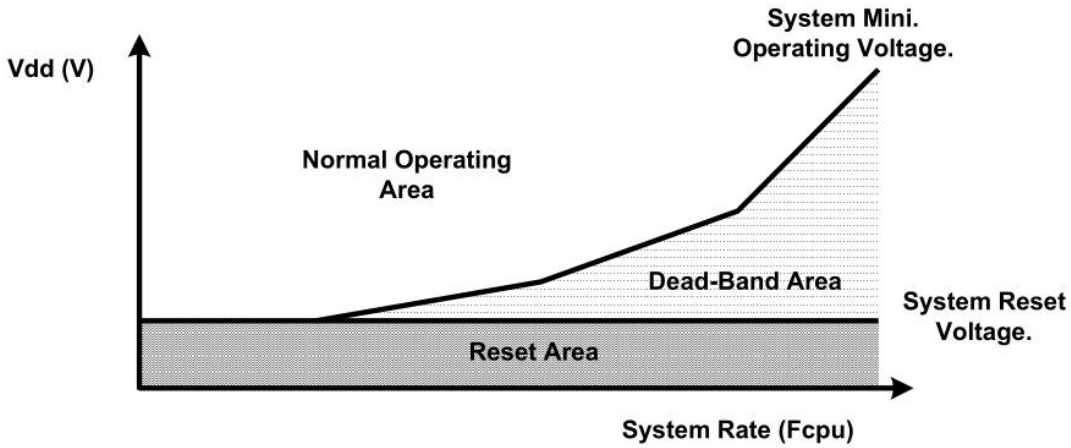
AC application:

In AC power application, the DC power is regulated from AC power source. This kind of power usually couples with AC noise that makes the DC power dirty. Or the external loading is very heavy, e.g. driving motor. The loading operating induces noise and overlaps with the DC power. VDD drops by the noise, and the system works under unstable power situation.

The power on duration and power down duration are longer in AC application. The system power on sequence protects the power on successful, but the power down situation is like DC low battery condition. When turn off the AC power, the VDD drops slowly and through the dead-band for a while.

3.7.5 THE SYSTEM OPERATING VOLTAGE

To improve the brown out reset needs to know the system minimum operating voltage which is depend on the system executing rate and power level. Different system executing rates have different system minimum operating voltage. The electrical characteristic section shows the system voltage to executing rate relationship.



Normally the system operation voltage area is higher than the system reset voltage to VDD, and the reset voltage is decided by LVD detect level. The system minimum operating voltage rises when the system executing rate upper even higher than system reset voltage. The dead-band definition is the system minimum operating voltage above the system reset voltage.

In order to avoid dead zone voltage, the corresponding LVR reset voltage point should be selected when the operating frequency is selected again.

The following table:

Frequency	LVR reset voltage point
Fcpu=8M	LVR=3.6V
Fcpu=4M	LVR=3.6V
Less than Fcpu=2M	-

Note:

1、When Fcpu=8M, JZ8P2612 must supply 5V and reset voltage must be 3.6V. And it can only be used in an environment no higher than 45 degrees.

2、When Fcpu=4M, JZ8P2612 recommend 3.6V reset voltage, which can be used in full temperature range. The minimum is 2.4V, but do not work in high temperature environment.

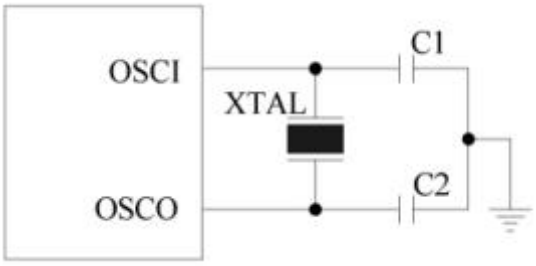
3.8 SYSTEM CLOCK

The JZ8P2612 has an integrated dual clock system, high speed clock and low speed clock. The high speed clock includes an internal high speed clock and an external high speed clock, selected by the compilation option. The low speed clock is provided by an internal low speed oscillator and controlled by the CLKMD bit of the OSCM register. Both high-speed clock and low-speed clock can be used as a system clock source.

Oscillator type	Description
IHRC Internal high speed oscillator	Most 32MHz
XT External crystal oscillator	4MHz/12MHz/32KHz
ILRC Internal low speed oscillator	16KHz(3V)/32KHz(5V)

3.8.1 EXTERNAL CRYSTAL OSCILLATOR

In most applications, pin OSCO and OSCI can be connected to crystals. The circuit diagram is shown below, applicable to both HXT and LXT modes. The table shows the recommended values for C1 and C2. Since the characteristics of each resonator are different, the user should select the appropriate values of C1 and C2 by referring to their specifications.



crystal oscillator circuit

Capacitance reference value:

Oscillator mode	Frequency patterns	Frequency	C1(pF)	C2(pF)
Crystal oscillator	LXT (32.768K)	32.768 KHz	40	40
	HXT (1M～16M)	1 MHz	30	30
		2 MHz	30	30
		4 MHz	20	20
		8 MHz	30	30

Note: the above data is for reference only, all subject to physical test.

3.8.2 INTERNAL RC OSCILLATOR MODE (IRC)

The internal RC oscillation mode frequency values are 32MHz, 16MHz, 8MHz, and 4MHz. IRC working frequency can be selected by setting the configuration bit of OPTION, and the corresponding relationship is as follows:

Firc	IRC frequency
32 M	IRC Frequency selective 32MHz
16 M	IRC Frequency selective 16MHz
8 M	IRC Frequency selective 8MHz
4 M	IRC Frequency selective 4MHz

JZ8P2612 provides a variety of frequency division options, which can be selected in OPTION, suitable for more occasions. The following table:

Clocks	Clocks Frequency division
8clock	Frequency division selection 8clock
16clock	Frequency division selection 16clock
32clock	Frequency division selection 32clock
64clock	Frequency division selection 64clock
128clock	Frequency division selection 128clock

3.8.3 OSCM MODE REGISTER

Bit 7	Bit6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	--	CPUM1	CPUM0	CLKMD	STPHX	--

Bit<4:3>:**CPUM<1:0>**-CPU operating mode control bits.

CPUS1	CPUS0	Working mode
0	0	Normal mode
0	1	Sleep mode
1	0	Green mode
1	1	Reserve

Bit<2>:**CLKMD**- System high/Low clock mode control bit.

0: Normal (dual) mode. System clock is high clock.

1: Slow mode. System clock is internal low clock.

Bit<1>:**STPHX**- External high-speed oscillator control bit.

0: Free run.

1: Stop. Internal low-speed RC oscillator is still running.

STPHX bit is the control bit of internal high speed RC oscillator and external high speed oscillator. When STPHX=0, internal high speed RC oscillator and external high speed oscillator operate normally. When STPHX=1, the external high speed oscillator and internal high speed RC oscillator stop operating. Different high speed clock options determine the different STPHX features.

IHRC_16M：STPHX=1, disable internal high speed RC oscillator;

RC, 4M, 12M, 32K：STPHX=1, disable external oscillator.

3.9 SYSTEM OPERATION MODE

3.9.1 OVERVIEW

JZ8P2612 can operate at four different clock frequencies in four working modes, These modes control oscillators, op-code operation and analog peripheral devices' operation.

1. Normal mode: System high-speed operating mode.
2. Slow mode: System low-speed operating mode.
3. Sleep mode: System sleep mode.
4. Green mode: System ideal mode.

Operating Mode	Normal Mode	Slow Mode	Green Mode	Sleep Mode
EHOSC	Running	By STPHX	By STPHX	Stop
IHRC	Running	By STPHX	By STPHX	Stop
ILRC	Running	Running	Running	Stop
CPU instruction	Executing	Executing	Stop	Stop
TC0	By TC0ENB	By TC0ENB	By TC0ENB Only PWM/Buzzer active.	Inactive
TC1	By TC1ENB	By TC1ENB	TBy TC1ENB Only PWM/Buzzer active.	Inactive
Watchdog timer	By Watch_Dog Code option	By Watch_Dog Code option	By Watch_Dog Code option	By Watch_Dog Code option
Internal interrupt	All active	All active	TC0	All inactive
External interrupt	All active	All active	All active	All inactive
Wakeup source	-	-	P0, TC0, Reset	P0, Reset

EHOSC: External high-speed oscillator (XIN/XOUT).
IHRC: Internal high-speed oscillator RC type.
ILRC: Internal low-speed oscillator RC type.

3.9.2 NORMAL MODE

The Normal Mode is system high clock operating mode. The system clock source is from high speed oscillator. The program is executed. After power on and any reset trigger released, the system inserts into normal mode to execute program. When the system is wake-up from power down mode, the system also inserts into normal mode. In normal mode, the high speed oscillator actives, and the power consumption is largest of all operating modes.

- *The program is executed, and full functions are controllable.*

- *The system rate is high speed.*
- *The high speed oscillator and internal low speed RC type oscillator active.*
- *Normal mode can be switched to other operating modes through OSCM register.*
- *Power down mode is wake-up to normal mode.*
- *Slow mode is switched to normal mode.*
- *Green mode from normal mode is wake-up to normal mode.*

3.9.3 SLOW MODE

The slow mode is system low clock operating mode. The system clock source is from internal low speed RC type oscillator. The slow mode is controlled by CLKMD bit of OSCM register. When CLKMD=0, the system is in normal mode. When CLKMD=1, the system inserts into slow mode. The high speed oscillator won't be disabled automatically after switching to slow mode, and must be disabled by SPTHX bit to reduce power consumption. In slow mode, the system rate is fixed Flosc/4 (Flosc is internal low speed RC type oscillator frequency).

- *The program is executed, and full functions are controllable.*
- *The system rate is low speed (Flosc/4).*
- *The internal low speed RC type oscillator actives, and the high speed oscillator is controlled by STPHX=1. In slow mode, to stop high speed oscillator is strongly recommendation.*
- *Slow mode can be switched to other operating modes through OSCM register.*
- *Sleep mode from slow mode is wake-up to normal mode.*
- *Normal mode is switched to slow mode.*
- *Green mode from slow mode is wake-up to slow mode.*

3.9.4 SLEEP MODE

The sleep mode is the system ideal status. No program execution and oscillator operation. Whole chip is under low power consumption status under 1uA. The sleep mode is waked up by P0 hardware level change trigger. Any operating modes into sleep mode, the system is waked up to normal mode. Inserting sleep mode is controlled by CPUM0 bit of OSCM register. When CPUM0=1, the system inserts into sleep mode. After system wake-up from sleep mode, the CPUM0 bit is disabled (zero status) automatically.

- *The program stops executing, and full functions are disabled.*
- *All oscillators including external high speed oscillator, internal high speed oscillator and internal low speed oscillator stop.*
- *The power consumption is under 1uA.*
- *The system inserts into normal mode after wake-up from power down mode.*
- *The power down mode wake-up source is P0 level change trigger.*

Note: If the system is in normal mode, to set STPHX=1 to disable the high clock oscillator. The system is under no system clock condition. This condition makes the system stay as power down mode, and can be wake-up by P0 level change trigger.

3.9.5 GREEN MODE

The green mode is another system ideal status not like power down mode. In power down mode, all functions and hardware devices are disabled. But in green mode, the system clock source keeps running, so the power consumption of green mode is larger than power down mode. In green mode, the program isn't executed, but the timer with wake-up function actives as enabled, and the timer clock source is the non-stop system clock. The green mode has 2 wake-up sources. One is the P0 level change trigger wake-up. The other one is internal timer with wake-up function occurring overflow. That's mean users can setup one fix period to timer, and the system is waked up until the time out. Inserting green mode is controlled by CPUM1 bit of OSCM register. When CPUM1=1, the system inserts into green mode. After system wake-up from green mode, the CPUM1 bit is disabled (zero status) automatically.

- *The program stops executing, and full functions are disabled.*
- *Only the timer with wake-up function actives.*
- *The oscillator to be the system clock source keeps running, and the other oscillators operation is depend on system operation mode configuration.*
- *If inserting green mode from normal mode, the system insets to normal mode after wake-up.*
- *If inserting green mode from slow mode, the system insets to slow mode after wake-up.*
- *The green mode wake-up sources are P0 level change trigger and unique time overflow.*
- *PWN and buzzer output functions active in green mode, but the timer can"t wake-up the system as overflow.*

Note: Sonix provides "GreenMode" macro to control green mode operation. It is necessary to use "GreenMode" macro to control system inserting green mode. The macro includes three instructions. Please take care the macro length as using BRANCH type instructions, e.g. bts0, bts1, b0bts0, b0bts1, ins, incms, decs, decms, cmprs, jmp, or the routine would be error.

3.9.6 OPERATING MODE CONTROL MACRO

Macro	Length	Description
SleepMode	1-word	The system insets into Sleep Mode
GreenMode	3-word	The system inserts into Green Mode.
SlowMode	2-word	The system inserts into Slow Mode and stops high speed oscillator.
Slow2Normal	5-word	The system returns to Normal Mode from Slow Mode. The macro includes operating mode switch, enable high speed oscillator, high speed oscillator warm-up delay time.

Example: Switch normal/slow mode to green mode and enable TC0 wake-up function.

```
; Set TC0 timer wakeup function.
        BOBCLR  FT0IEN      ; To disable TC0 interrupt service
        BOBCLR  FT0ENB      ; To disable TC0 timer
        MOV     A,#20H
        BOMOV   T0M,A       ; To set TC0 clock = Fcpu / 64
        MOV     A,#64H
        BOMOV   T0C,A       ; To set T0C initial value = 64H (To set TC0 interval
                            ; = 10 ms)
        BOBCLR  FT0IEN      ; To disable TC0 interrupt service
        BOBCLR  FT0IRQ      ; To clear TC0 interrupt request
        BOBSET  FT0ENB      ; To enable TC0 timer
; Go into green mode
        GreenMode           ; Declare "GreenMode" macro directly.
```

3.9.7 WAKEUP

Under sleep mode or green mode, program doesn't execute. The wakeup trigger can wake the system up to normal mode or slow mode. The wakeup trigger sources are external trigger (P0 level change) and internal trigger (TC0 timer overflow).

When the system is in sleep mode, the high clock oscillator stops. When waked up from sleep mode, MCU waits for 2048 external high-speed oscillator clocks and 8 internal high-speed oscillator clocks as the wakeup time to stable the oscillator circuit. After the wakeup time, the system goes into the normal mode.

Note: Wakeup from green mode is no wakeup time because the clock doesn't stop in green mode.

The value of the external high clock oscillator wakeup time is as the following:

The Wakeup time = 1/Fhosc * 2048 (sec) + high clock start-up time

Example: Insleep mode, the system is waked up. After the wakeup time, the system goes into normal mode. The wakeup time is as the following:

The wakeup time = 1/Fhosc * 2048 = 0.512 ms (Fhosc = 4MHz)

The total wakeup time = 0.512 ms + oscillator start-up time

The value of the internal high clock oscillator RC type wakeup time is as the following.

The wakeup time = 1/Fhosc * 8 (sec) + high clock start-up time

Example: In sleep mode, the system is waked up. After the wakeup time, the system goes into normal mode. The wakeup time is as the following:

The wakeup time = 1/Fhosc *8 = 0.5 us (Fhosc = 16MHz)

Note: The high clock start-up time is depended on the VDD and oscillator type of high clock.

4 CODE OPTION REGISTER

CODE OPTION	Options	Description
Code protect	Enable protect	Burn mode data protect
	Disable protect	Burn mode data is not protect
Watch dog	Enable	Watch dog enable
	Always On	Watch dog always On
	Disable	Disable Watch dog
Oscillation mode	HIRC	Internal high speed oscillator
	HIRC-RTC	Internal high speed oscillator and real timer counter
	LXT	External low speed crystal
	HXT	External high speed crystal
HIRC frequency	4M	Internal 4M frequency
	8M	Internal 8M frequency
	16M	Internal 16M frequency
	32M	Internal 32M frequency
Clocks Frequency division	8 Clocks	Clock frequency division select 8 Clocks
	16 Clocks	Clock frequency division select 16 Clocks
	32 Clocks	Clock frequency division select 32 Clocks
	64 Clocks	Clock frequency division select 64 Clocks
	128 Clocks	Clock frequency division select 128 Clocks
System filtering	Enable	Enable system filtering
	Disable	Disable system filtering
Reset port pull up	Enable	Enable P04 port pull up
	Disable	Disable P04 port pull up
P04 port	As GPI	P04 port as General Purpose Input
	As RST	P04 port as reset port
RST port Reset the time delay	No delay	RST port without delay
	Delay 128 CLK	RST port delay 128 CLK
P04 port Open drain output	Enable	P04 port enable Open drain output
	Disable	P04 port disable Open drain output
LVD	LVR-L	LVR:2.0V
	LVR-M	LVR:2.0V LVD:2.4V
	LVR-H	LVR:2.4V LVD:3.6V
	LVD-S	LVR:3.6V

5 ELECTRICAL CHARACTERISTIC

5.1 ABSOLUTE MAXIMUM RATING

Operating ambient temperature (℃):	(√) E:-40-85;
Storage ambient temperature (℃):	(√) -65～+150;
Supply voltage (V)	(√) -0.3～6;
Input in voltage (V)	(√) GND-0.3～VDD+1;
Output in voltage (V)	(√) GND-0.3～VDD+1;

5.2 DC CHARACTERISTIC

(T=25℃, VDD=5±5%V,GND=0V)

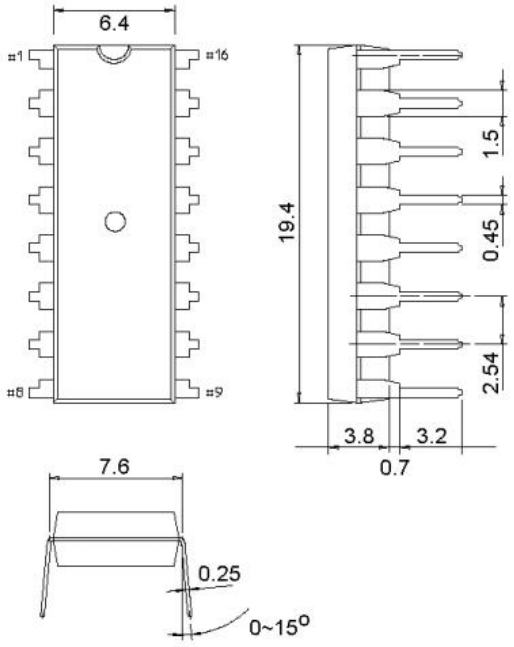
SYMBOL	PARAMETER	DESCRIPTION	MIN	TYP	MAX	UNIT
IRC1	IRC1 (after calibration)	OPTION select 4MHz	-	4	-	MHz
IRC2	IRC2 (after calibration)	OPTION select 8MHz	-	8	-	MHz
IRC3	IRC3 (after calibration)	OPTION select 16MHz	-	16	-	MHz
IRC4	IRC4 (after calibration)	OPTION select 32MHz	-	32	-	MHz
IOH1	Output high level drive (in addition to P04)	Ioh=4.4V	16	17	18	mA
IOL1	Output low level drive	Iol=0.6V	19	20	21	mA
IPH	Pull-up current	Enable pull-up, input connect gnd	55	58	60	μA
Isb1	Green mode current 1	Low speed mode to green mode	1.8	2	3	μA
Isb2	Green mode current 2	Normal mode(16M)to green mode	280	290	300	μA
Isb3	Green mode current 3	Normal mode(4M)to green mode	110	120	130	μA
Ilb1	Low speed mode current	Low speed mode to green mode	4	5	6	μA
ILb1	Sleep mode current 1	Normal sleep	-	-	1	μA
ILb2	Sleep mode current 2	Enable LVD sleep	4	5	6	μA
Iop1	Working current1 (VDD=5V)	IRC=32MHz 8clock	-	1.6	-	mA
Iop2	Working current 2(VDD=5V)	IRC=16MHz 8clock	-	0.9	-	mA
Iop3	Working current 3(VDD=5V)	IRC=16MHz 16clock	-	0.6	-	mA
Iop4	Working current 4(VDD=5V)	IRC=8MHz 8clock	-	0.5	-	mA
Iop5	Working current 5(VDD=5V)	IRC=4MHz 8clock	-	0.3	-	mA
Iop6	Working current 6(VDD=5V)	XT=12MHz 8clock	-	1	-	mA
Iop7	Working current 7(VDD=5V)	XT=4MHz 8clock	-	0.6	-	mA
Iop8	Working current 8(VDD=5V)	XT=32.768KHz 2clock	-	13	-	μA

LVR	LVR Voltage	Select LVR reset voltage	Vlvr-0.2	Vlvr	Vlvr+	V
LVD	LVD Voltage	Select LVD voltage is Vlvd	Vlvd-0.2	Vlvd	Vlvd+	V

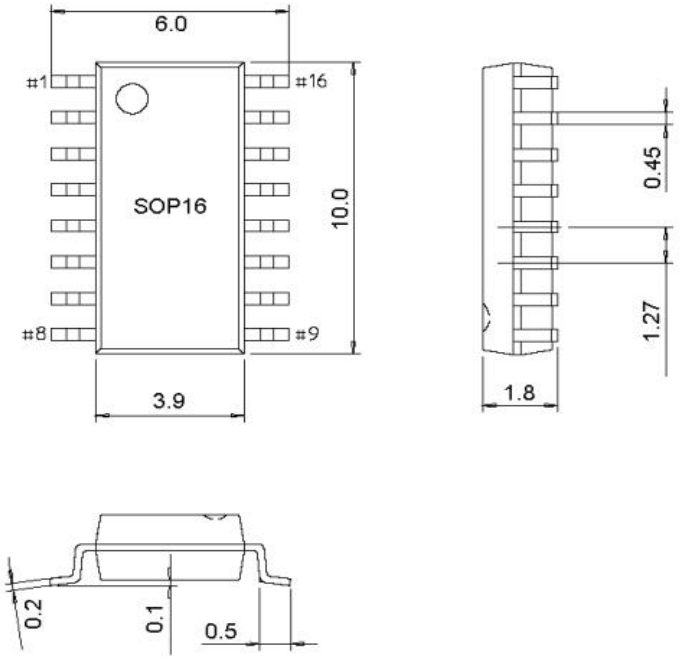
6 PACKAGE INFORMATION

6.1 16PIN PACKAGE INFORMATION

Unit (mm)



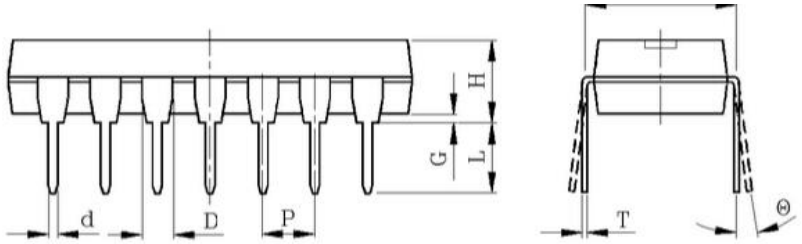
DIP16 Package

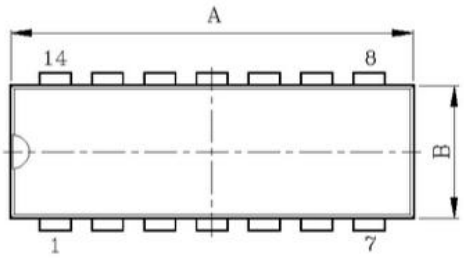


SOP16 Package

6.2 14PIN PACKAGE INFORMATION

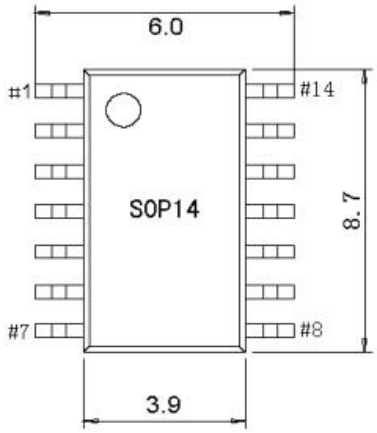
Unit (mm)

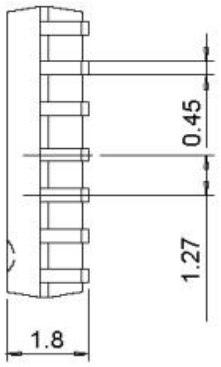


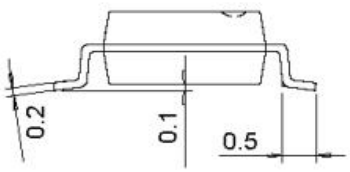


DIM	MILLIMETERS
A	19.3±0.2
B	6.45±0.2
D	1.52±0.1
d	0.46±0.1
G	0.50 MIN
H	3.8±0.3
L	3.3±0.3
P	2.54
T	0.25+0.1/−0.05
W	7.62
Θ	0 ~ 15°

DIP14 Package



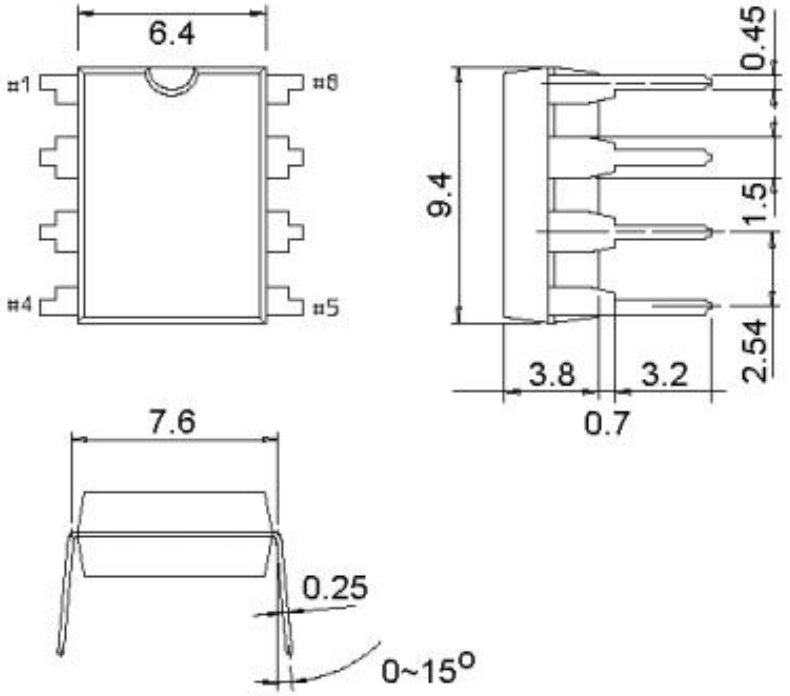




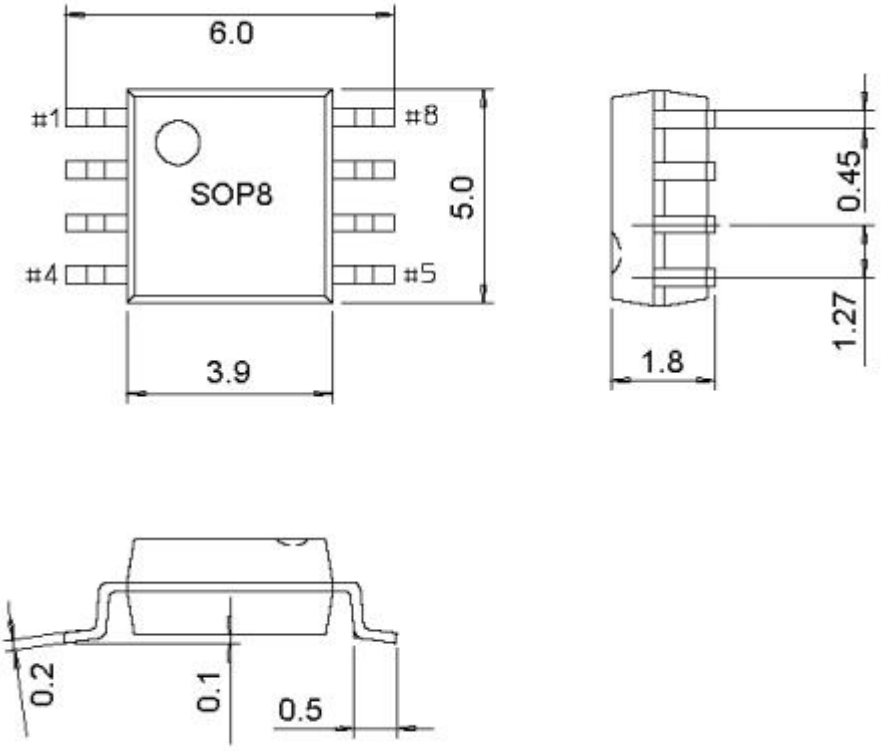
SOP14 Package

6.3 8PIN PACKAGE INFORMATION

Unit (mm)



DIP8 Package



SOP8 Package

6.4 6PIN PACKAGE INFORMATION

Unit (mm)

Size / Mark	Minimum	Maximum	Size / Mark	Minimum	Maximum
A	2.82	3.02	C	1.05	1.15
e	0.95 (BSC)		C1	0.03	0.15
b	0.28	0.45	C2	0.12	0.23
B	1.50	1.70	L	0.35	0.55
B1	2.75	3.05	θ	0°	8°

