SN8PC01

USER'S MANUAL

Preliminary Specification

SONiX 8-Bit REMOTE ASIC

AMENDENT HISTORY

Version	Date	Description
VER 1.0	Dec. 2003	Preliminary Version 1.0
VER 1.1	Mar. 2004	Modify P2.4, P2.5 from output only to bi-direction I/O. Add P2M register. Y register changes from 8-bit to 4-bit.
VER 1.2	Mar. 2005	1. XIN capacitor is 68pF, XOUT capacitor is 20pF. 2. After power on reset and external reset, P5.4 is logic high status.
VER 1.3	Aug. 2005	Add IR output component circuit and development tool chapter.
VER 1.4	Sep. 2005	1. Modify minimum operating voltage from 2.0V to 2.2V. 2. Update IDE version to SN8PC01IDE_V101.

Table of Content

12 PACKAGE INFORMATION 42

1 PRODUCT OVERVIEW

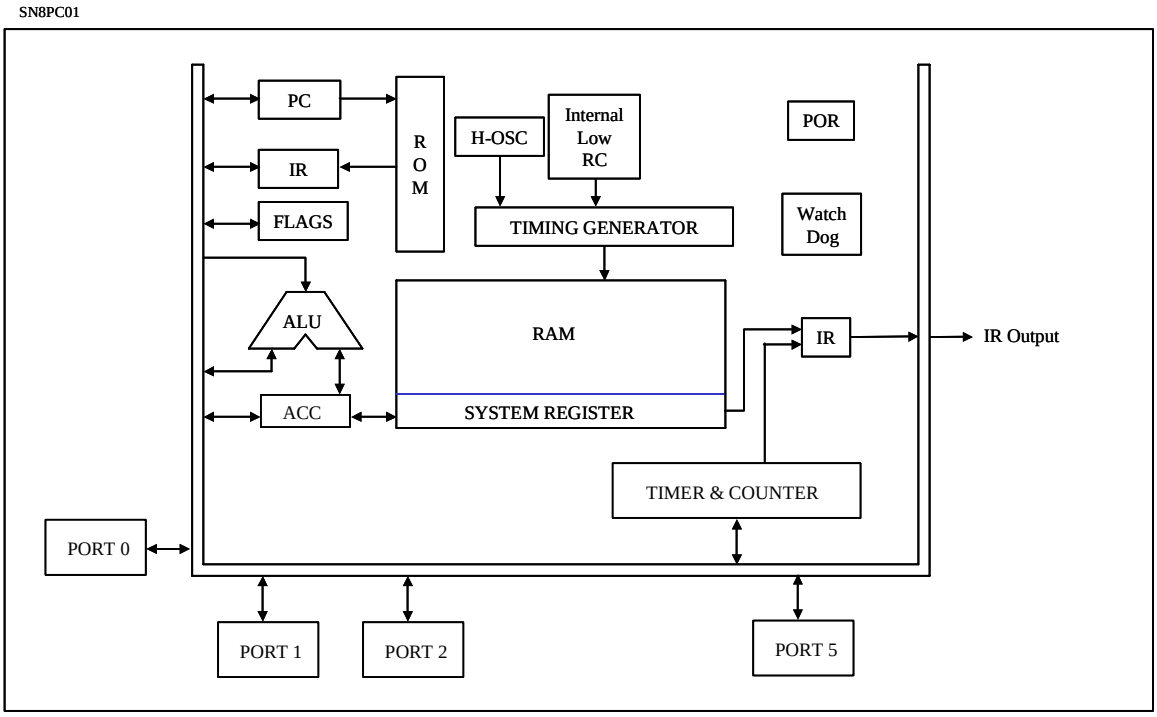
1.1 FEATURES

- **Memory configuration**
  OTP ROM size: 0.5K * 16 bits.
  RAM size: 32 * 8 bits.
- **4 levels stack buffer.**
- **I/O pin configuration**
  Input only: P0, P1
  Output only: P2, P5
  I/O: P2.4, P2.5
  Wakeup: P0, P1
  Pull-up resisters: P1
- **One channel IR output.**
- **On chip watchdog timer.**
- **System clocks**
  External high clock: Crystal type 455KHz
- **Operating modes**
  Normal mode: High clock active
- **Package (Chip form support)**
  PDIP 20 pins
  SOP 20 pins
- **Powerful instructions**
  4T Instruction cycle.
  Instruction's length is one word.
  Most of instructions are one cycle only.
  Maximum instruction cycle is two.
  All ROM area JMP instruction.
  All ROM area lookup table function (MOVC)

1.1.1 SELECTION TABLE

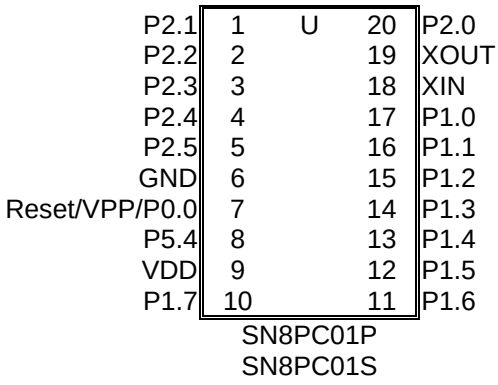
CHIP	ROM	RAM	Stack	I/O	IR	Wakeup Pin No.	Package
SN8PC01	0.5K*16	32	4	16	V	9	DIP20/SOP20

1.2 SYSTEM BLOCK DIAGRAM



1.3 PIN ASSIGNMENT

SN8PC01P (P-DIP 20 pins)
SN8PC01S (SOP 20 pins)



SN8PC01P
SN8PC01S

1.4 PIN DESCRIPTIONS

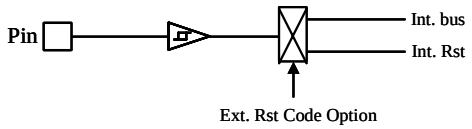
➢ ***SN8PC01***

PIN NAME	*TYPE*	*DESCRIPTION*
VDD, VSS	P	Power supply input pins for digital circuit.
RST/VPP/P0.0	I/P	System external reset input pin. Schmitt trigger structure, active “low”, normal stay to “high”. OTP 12.3V power input pin in programming mode. Port 0.0 input only pin with Schmitt trigger structure and **no pull-up resistor**. Note: This pin is with metal option. Default is VPP/RST. P0.0 circuit must be with pull-up resistor, or the system won’t be stable as VPP/RST condition.
XIN	I	455KHz oscillator input pin. The capacitor must be **68pF**.
XOUT	O	455KHz oscillator output pin. The capacitor must be **20pF**.
P1.0 ~P1.7	I	Port 1.0~P1.7 input only pin with Schmitt trigger structure and pull-up resistor.
P2.0~P2.3	O	Port 2.0~P2.5 output only pin.
P2.4~P2.5	I/O	Port 2.4, P2.5 bi-direction I/O pin with Schmitt trigger structure and pull-up resistor.
P5.4	O	Port 5.4 is 38KHz signal output controlled by TC0OUT and P54 bits. **After reset, P5.4 outputs “high”.** **TC0OUT=1 and P54=1, P5.4 keeps high status.** **TC0OUT=1 and P54=0, P5.4 outputs 38KHz signal for IR.** **TC0OUT=0, P5.4 is general output pin.**

1.5 PIN CIRCUIT DIAGRAMS

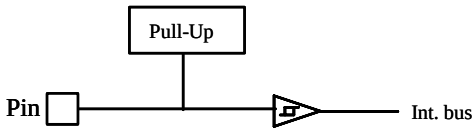
Port 0.0 structure:

Port 0.0 structure



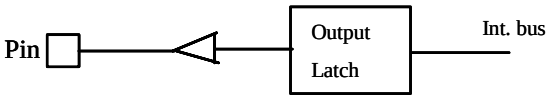
Port 1 structure:

Port 1 structure

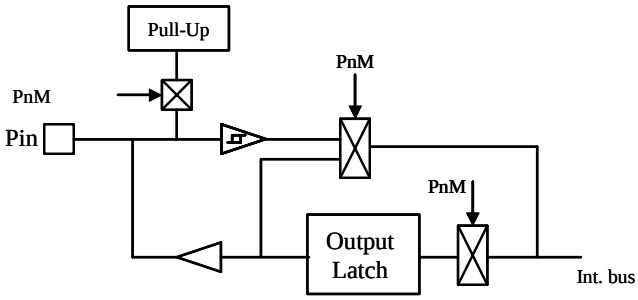


Port 2.0~P2.3 structure:

Port 2 structure

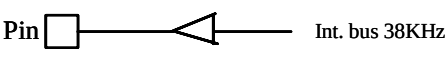


Port 2.4~P2.5 structure:



Port 5.4 structure:

Port 5.4 structure



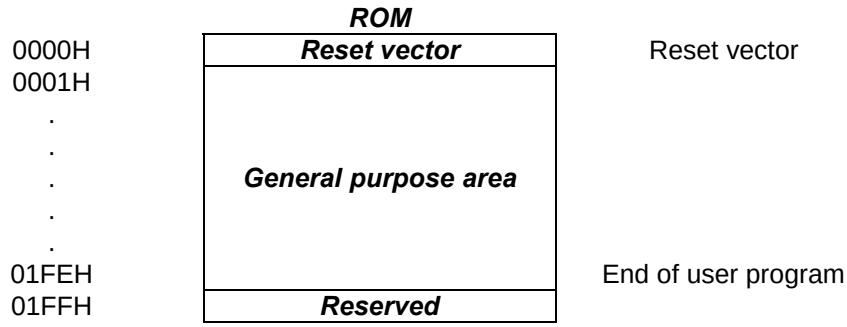
➢ ***Note: All of the latch output circuits are push-pull structures.***

2 CENTRAL PROCESSOR UNIT (CPU)

2.1 MEMORY MAP

2.1.1 PROGRAM MEMORY (ROM)

➢ ***0.5K words ROM***



USER RESET VECTOR ADDRESS (0000H)

A 1-word vector address area is used to execute system reset. After power on reset or watchdog timer overflow reset, then the chip will restart the program from address 0000h and all system registers will be set as default values. The following example shows the way to define the reset vector in the program memory.

➲ **Programming Tip: Defining Reset Vector**

```
CHIP SN8PC01
        ORG     0
START:                          ; 0000H. Reset vector and the head of user program.
        .                       ; User program
        .
        .
        .
        ENDP                    ; End of program
```

GENERAL PURPOSE PROGRAM MEMORY AREA

The ROM location 0001H~01FEH are used as general-purpose memory. The area is to store both instruction's op-code and look-up table data. The SN8PC01 includes jump table function by using program counter (PC) and look-up table function by using ROM code registers (R, Y, Z).

The boundary of program memory is separated by the high-byte program counter (PCH) every 100H. In jump table function and look-up table function, the program counter can't leap over the boundary by program counter automatically. Users need to modify the PCH value to "PCH+1" when the PCL overflows (from 0FFH to 000H).

LOOK-UP TABLE DESCRIPTION

In the ROM's data lookup function, Y register is pointed to the bit 8~bit 15 and Z register to the bit 0~bit 7 data of ROM address. After MOVC instruction executed, the low-byte data will be stored in ACC and high-byte data stored in R register.

➲ **Example: To look up the ROM data located "TABLE1".**

```
                MOV     A #TABLE1$M     ; To set lookup table1's middle address
                MOV     Y, A
                MOV     A #TABLE1$L     ; To set lookup table1's low address.
                MOV     Z, A
                MOVC                    ; To lookup data, R = 00H, ACC = 35H
                                        ;
                                        ; Increment the index address for next address
                INCMS   Z               ; Z+1
                JMP     @F              ; Not overflow
                INCMS   Y               ; Z overflow (FFH → 00), → Y=Y+1
                NOP                     ;
                                        ;
@@:             MOVC                    ; To lookup data, R = 51H, ACC = 05H.
                .       .               ;
TABLE1:         DW      0035H           ; To define a word (16 bits) data.
                DW      5105H           ; "
                DW      2012H           ; "
```

➢ **CAUSION: The Y register will not increase automatically when Z register crosses boundary from 0xFF to 0x00. Therefore, user must take care such situation to avoid loop-up table errors. If Z register overflows, Y register must be added one. The following INC_YZ macro shows a simple method to process Y and Z registers automatically.**

➢ ***Note: Because the program counter (PC) is only 10-bit, the X register is useless in the application. Users can omit "B0MOV X, #TABLE1$H". SONiX ICE supports larger program memory addressing capability. Please be sure that X register is "0" to avoid unpredicted error in loop-up table operation.***

➲ **Example: INC_YZ Macro**

```
INC_YZ          MACRO
                INCMS   Z               ; Z+1
                JMP     @F              ; Not overflow

                INCMS   Y               ; Y+1
                NOP                     ; Not overflow
@@:
                ENDM
```

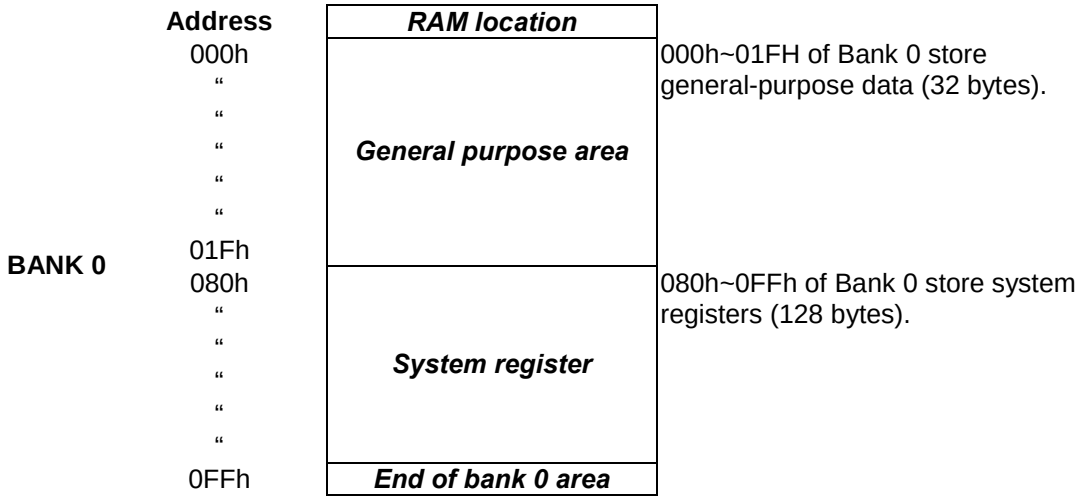
The other example of loop-up table is to add Y or Z index register by accumulator. Please be careful if “carry” happen.

➲ **Example: Increase Y and Z register by B0ADD/ADD instruction**

```
        MOV     A #TABLE1$M     ; To set lookup table1's middle address
        MOV     Y, A
        MOV     A #TABLE1$L     ; To set lookup table1's low address.
        MOV     Z, A

        MOV     A, BUF          ; Z = Z + BUF.
        ADC     Z, A

        BTS1    FC              ; Check the carry flag.
        JMP     GETDATA         ; FC = 0
        INCMS   Y               ; FC = 1. Y+1.
        NOP

GETDATA:                        ;
        MOVC                    ; To lookup data. If BUF = 0, data is 0x0035
                                ; If BUF = 1, data is 0x5105
                                ; If BUF = 2, data is 0x2012
                                .
                                .
        .       .               ;
TABLE1: DW      0035H           ; To define a word (16 bits) data.
        DW      5105H           ; “
        DW      2012H           ; “
```

2.1.2 DATA MEMORY (RAM)

- 32 * 8-bit RAM



2.1.3 CODE OPTION TABLE

Code Option	Content	Function Description
Watch_Dog	Enable	Enable Watchdog function.
	Disable	Disable Watchdog function.
Reset_Pin	P00	The pin is to be input only pin P0.0 without pull-up resister.
	Reset	The pin is to be external reset pin.
Security	Enable	Enable ROM code Security function.
	Disable	Disable ROM code Security function.

2.1.4 SYSTEM REGISTER

BYTES of SYSTEM REGISTER

➢ ***SN8PC01***

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	-	-	R	Z	Y	-	PFLAG	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C	-	-	P2M	-	-	-	-	-	-	-	OSCM	-	-	-	PCL	PCH
D	P0	P1	P2	-	-	P5	-	-	-	-	TC0M	-	-	-	-	STKP
E	-	-	-	-	-	-	-	@YZ	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	STK3L	STK3H	STK2L	STK2H	STK1L	STK1H	STK0L	STK0H

Description

PFLAG = Special flag register.
OSCM = Oscillator mode register.
TC0M = TC0OUT control register.
STKP = Stack pointer buffer.
@YZ = RAM YZ indirect addressing index pointer.
R = Working register and ROM look-up data buffer.
Y, Z = Working, @YZ and ROM addressing register.
Pn = Port n data buffer.
PnM = Port n mode buffer.
PCH, PCL = Program counter.
STK0~STK3 = Stack 0 ~ stack 3 buffer.

BITS of SYSTEM REGISTER

➢ ***SN8PC01 system register table***

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	-	-	-	-	YBIT3	YBIT2	YBIT1	YBIT0	R/W	Y
086H	-	-	-	-	-	C	-	Z	R/W	PFLAG
0C2H	-	-	P25M	P24M	-	-	-	-	W	P2M
0CAH	0	WDRST	-	-	CPUM0	-	-	-	R/W	OSCM
0CEH	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	R/W	PCL
0CFH	-	-	-	-	-	-	-	PC8	R/W	PCH
0D0H	-	-	-	-	-	-	-	P00	R	P0 data buffer
0D1H	P17	P16	P15	P14	P13	P12	P11	P10	R	P1 data buffer
0D2H	-	-	P25	P24	P23	P22	P21	P20	R/W	P2 data buffer
0D5H	-	-	-	P54	-	-	-	-	W	P5 data buffer
0DAH	-	-	-	-	-	-	TC0OUT	-	R/W	TC0M
0DFH	-	-	-	-	-	STKPB2	STKPB1	STKPB0	R/W	STKP stack pointer
0E7H	@YZ7	@YZ6	@YZ5	@YZ4	@YZ3	@YZ2	@YZ1	@YZ0	R/W	@YZ index pointer
0F8H	S3PC7	S3PC6	S3PC5	S3PC4	S3PC3	S3PC2	S3PC1	S3PC0	R/W	STK3L
0F9H	-	-	-	-	-	-	-	S3PC8	R/W	STK3H
0FAH	S2PC7	S2PC6	S2PC5	S2PC4	S2PC3	S2PC2	S2PC1	S2PC0	R/W	STK2L
0FBH	-	-	-	-	-	-	-	S2PC8	R/W	STK2H
0FCH	S1PC7	S1PC6	S1PC5	S1PC4	S1PC3	S1PC2	S1PC1	S1PC0	R/W	STK1L
0FDH	-	-	-	-	-	-	-	S1PC8	R/W	STK1H
0FEH	S0PC7	S0PC6	S0PC5	S0PC4	S0PC3	S0PC2	S0PC1	S0PC0	R/W	STK0L
0FFH	-	-	-	-	-	-	-	S0PC8	R/W	STK0H

➢ ***Note***

a): To avoid system error, please be sure to put all the "0" and "1" as it indicates in the above table
b). All of register names had been declared in SN8ASM assembler.
c). One-bit name had been declared in SN8ASM assembler with "F" prefix code.
d)."bset", "bclr" instructions are only available to the "R/W" registers.
e). For detail description, please refer to the "System Register Quick Reference Table"

Y, Z REGISTERS

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

- can be used as general working registers
- can be used as RAM data pointers with @YZ register
- can be used as ROM data pointer with the MOVC instruction for look-up table

084H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Y	-	-	-	-	YBIT3	YBIT2	YBIT1	YBIT0
Read/Write	-	-	-	-	R/W	R/W	R/W	R/W
After reset	-	-	-	-	0	0	0	0

083H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Z	ZBIT7	ZBIT6	ZBIT5	ZBIT4	ZBIT3	ZBIT2	ZBIT1	ZBIT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

➲ **Example: uses YZ register as the data pointer to access data in the RAM address 025H of bank0.**

```
MOV     A #00H      ; To set RAM bank 0 for Y register
MOV     Y, A
MOV     A #25H      ; To set location 25H for Z register
MOV     Z, A
MOV     A, @YZ      ; To read a data into ACC
```

➲ **Example: uses the YZ register as data pointer to clear the RAM data**

```
        MOV     A #00H      ; To set RAM bank 0 for Y register
        MOV     Y, A
        MOV     A #25H      ; To set location 25H for Z register
        MOV     Z, A

CLR_YZ_BUF:
        MOV     A, #0       ; Clear @YZ to be zero
        MOV     @YZ, A

        DECMS   Z           ; Z – 1, if Z= 0, finish the routine
        JMP     CLR_YZ_BUF  ; Not zero

        MOV     A, #0
        MOV     @YZ, A
END_CLR:                    ; End of clear general purpose data memory area of bank 0
        .
```

R REGISTERS

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

- can be used as working register
- 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).

082H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R	RBIT7	RBIT6	RBIT5	RBIT4	RBIT3	RBIT2	RBIT1	RBIT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

➢ ***Note: Please refer to the "LOOK-UP TABLE DESCRIPTION" about R register look-up table application.***

2.2 ACCUMULATOR

The ACC is an 8-bit data register responsible for transferring or manipulating data between ALU and data memory. If the result of operating is zero (Z) or carry (C) occurrence, then these flags will be set to PFLAG register.

ACC is not in data memory (RAM), so ACC is access by “MOV” instruction during the instant addressing mode.

➲ **Example: Read and write ACC value.**

```
; Read ACC data and store in BUF data memory
                MOV         BUF, A
; Write a immediate data into ACC
                MOV         A, #0FH
; Write ACC data from BUF data memory
                MOV         A, BUF
```

2.3 PROGRAM FLAG

The PFLAG includes carry flag (C) and zero flag (Z). If the result of operating is zero or there is carry occurrence, then these flags will be set to PFLAG register.

086H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	-	-	-	-	-	C	-	Z
Read/Write	-	-	-	-	-	R/W	-	R/W
After reset	-	-	-	-	-	0	-	0

2.3.1 CARRY FLAG

C = 1: When executed arithmetic addition with overflow or executed arithmetic subtraction without borrow or executed rotation instruction with logic “1” shifting out.
C = 0: When executed arithmetic addition without overflow or executed arithmetic subtraction with borrow or executed rotation instruction with logic “0” shifting out.

2.3.2 ZERO FLAG

Z = 1: When the content of ACC or target memory is zero after executing instructions involving a zero flag.
Z = 0: When the content of ACC or target memory is not zero after executing instructions involving a zero flag.

2.4 PROGRAM COUNTER

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.

SN8PC01

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC	-	-	-	-	-	-	-	PC8	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
After reset	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0
	PCH								PCL							

2.4.1 ONE ADDRESS SKIPPING

There are nine instructions (CMPRS, INCMS, DECMS, BTS0, BTS1) with one address skipping function. If the result of these instructions is true, the PC will add 2 steps to skip next instruction.

If the condition of bit test instruction is true, the PC will add 2 steps to skip next instruction.

```
            BTS1    FC          ; To skip, if Carry flag = 1
            JMP     C0STEP      ; Else jump to C0STEP.
            .
C0STEP:     NOP

            MOV     A, BUF0     ; Move BUF0 value to ACC.
            BTS0    FZ          ; To skip, if Zero flag = 0.
            JMP     C1STEP      ; Else jump to C1STEP.
            .
C1STEP:     NOP
```

If the ACC is equal to the immediate data or memory, the PC will add 2 steps to skip next instruction.

```
            CMPRS   A, #12H     ; To skip, if ACC = 12H.
            JMP     C0STEP      ; Else jump to C0STEP.
            .
C0STEP:     NOP
```

If the destination increased by 1, which results overflow of 0xFF to 0x00, the PC will add 2 steps to skip next instruction.

INCMS instruction:

```
                INCMS     BUF0
                JMP       C0STEP        ; Jump to C0STEP if BUF0 is not zero.
                …
C0STEP:         NOP
```

If the destination decreased by 1, which results underflow of 0x00 to 0xFF, the PC will add 2 steps to skip next instruction.

DECMS instruction:

```
                DECMS     BUF0
                JMP       C0STEP        ; Jump to C0STEP if BUF0 is not zero.
                …
C0STEP:         NOP
```

2.4.2 MULTI-ADDRESS JUMPING

Users can jump around the multi-address by either JMP instruction or ADC M, An instruction (M = PCL) to activate multi-address jumping function. If carry flag occurs after execution of ADC A,PCL the carry flag will not affect PCH register.

➲ **Example: If PC = 0123H (PCH = 01H、PCL = 23H)**

```
; PC = 0123H
                MOV       A, #28H
                MOV       PCL, A        ; Jump to address 0128H
                .           .
                .           .
; PC = 0128H    .           .
                MOV       A, #00H
                MOV       PCL, A        ; Jump to address 0100H
```

2.5 ADDRESSING MODE

2.5.1 IMMEDIATE ADDRESSING MODE

The immediate addressing mode uses an immediate data to set up the location (“ MOV A, # I ”) in ACC or specific RAM.

Immediate addressing mode

```
MOV     A, #12H     ; To set an immediate data 12H into ACC
```

2.5.2 DIRECTLY ADDRESSING MODE

The directly addressing mode moves the content of RAM location in or out of ACC.(“ MOV A,12H ”, “ MOV 12H, A ”).

Directly addressing mode

```
MOV     A, 12H      ; To get a content of location 12H of bank 0 and save in ACC
```

2.5.3 INDIRECTLY ADDRESSING MODE

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

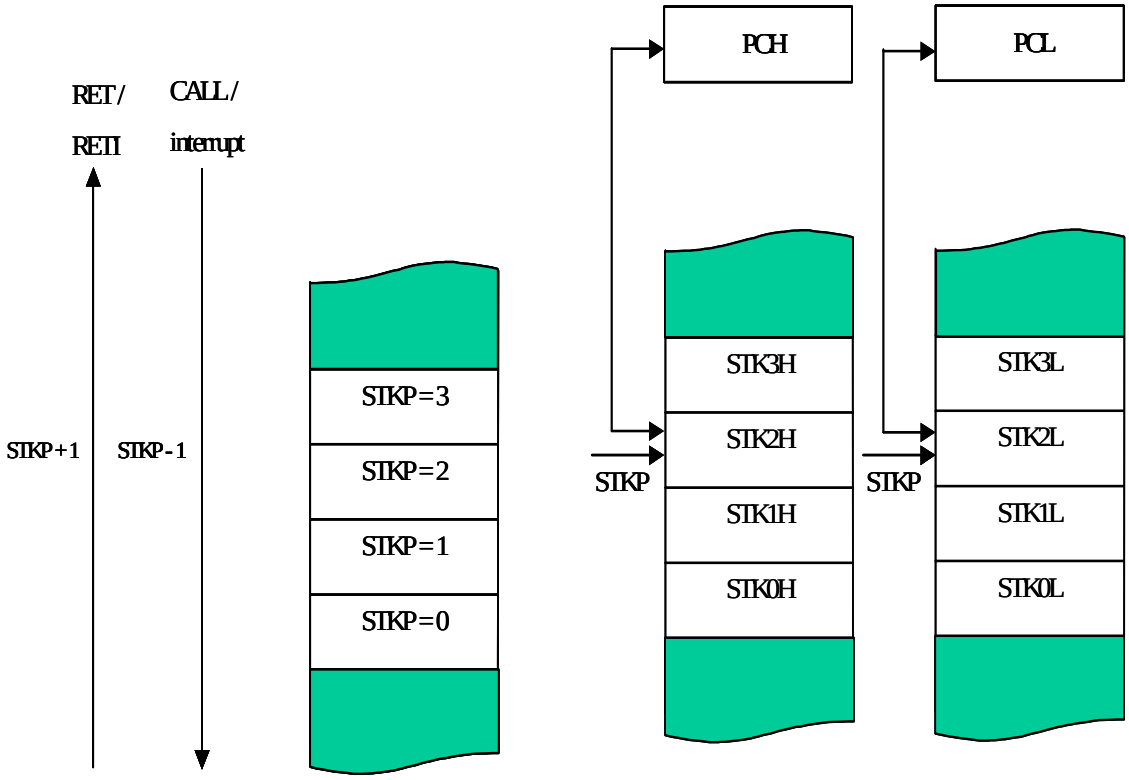
➲ **Example: Indirectly addressing mode with @YZ register**

```
MOV     A, #1       ; To clear Y register to access RAM bank 0.
MOV     Y, A
MOV     A, #12H     ; To set an immediate data 12H into Z register.
MOV     Z, A
MOV     A, @YZ      ; Use data pointer @YZ reads a data from RAM location
                    ; 012H into ACC.
```

2.6 STACK OPERATIONS

2.6.1 OVERVIEW

The stack buffer of SN8PC01 has 4-level. These buffers are designed to push and pop up program counter's (PC) data when interrupt service routine is executed. The STKP register is a pointer designed to point active level in order to push or pop up data from stack buffer. The STKnH and STKnL are the stack buffers to store program counter (PC) data.



2.6.2 STACK REGISTERS

The stack pointer (STKP) is a 3-bit register to store the address used to access the stack buffer, 10-bit data memory (STKnH and STKnL) set aside for temporary storage of stack addresses.

The two stack operations are writing to the top of the stack (push) and reading from the top of stack (pop). Push operation decrements the STKP and the pop operation increments each time. That makes the STKP always point to the top address of stack buffer and write the last program counter value (PC) into the stack buffer.

The program counter (PC) value is stored in the stack buffer before a CALL instruction executed or during interrupt service routine. Stack operation is a LIFO type (Last in and first out). The stack pointer (STKP) and stack buffer (STKnH and STKnL) are located in the system register area bank 0.

➢ ***SN8PC01***

0DFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKP	-	-	-	-	-	STKPB2	STKPB1	STKPB0
Read/Write	-	-	-	-	-	R/W	R/W	R/W
After reset	-	-	-	-	-	1	1	1

STKPBn: Stack pointer (n = 0 ~ 2)

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

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

➢ ***SN8PC01***

0F0H~0FFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKnH	-	-	-	-	-	-	SnPC9	SnPC8
Read/Write	-	-	-	-	-	-	R/W	R/W
After reset	-	-	-	-	-	-	0	0

STKn = <STKnH , STKnL> (n = 3 ~ 0)

➢ ***SN8PC01***

0F0H~0FFH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STKnL	SnPC7	SnPC6	SnPC5	SnPC4	SnPC3	SnPC2	SnPC1	SnPC0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

For SN8PC01 : STKn = <STKnH , STKnL> (n = 3 ~ 0)

2.6.3 STACK OPERATION EXAMPLE

The two kinds of Stack-Save operations refer to the stack pointer (STKP) and write the content of program counter (PC) to the stack buffer are CALL instruction and interrupt service. Under each condition, the STKP decreases and points to the next available stack location. The stack buffer stores the program counter about the op-code address. The Stack-Save operation is as the following table.

➢ ***SN8PC01***

Stack Level	***STKP Register***			***Stack Buffer***		***Description***
	STKPB2	**STKPB1**	**STKPB0**	**High Byte**	**Low Byte**	
0	1	1	1	Free	Free	-
1	1	1	0	STK0H	STK0L	-
2	1	0	1	STK1H	STK1L	-
3	1	0	0	STK2H	STK2L	-
4	0	1	1	STK3H	STK3L	-
> 4	0	1	0	-	-	**Stack Over, error**

There are Stack-Restore operations correspond to each push operation to restore the program counter (PC). The RETI instruction uses for interrupt service routine. The RET instruction is for CALL instruction. When a pop operation occurs, the STKP is incremented and points to the next free stack location. The stack buffer restores the last program counter (PC) to the program counter registers. The Stack-Restore operation is as the following table.

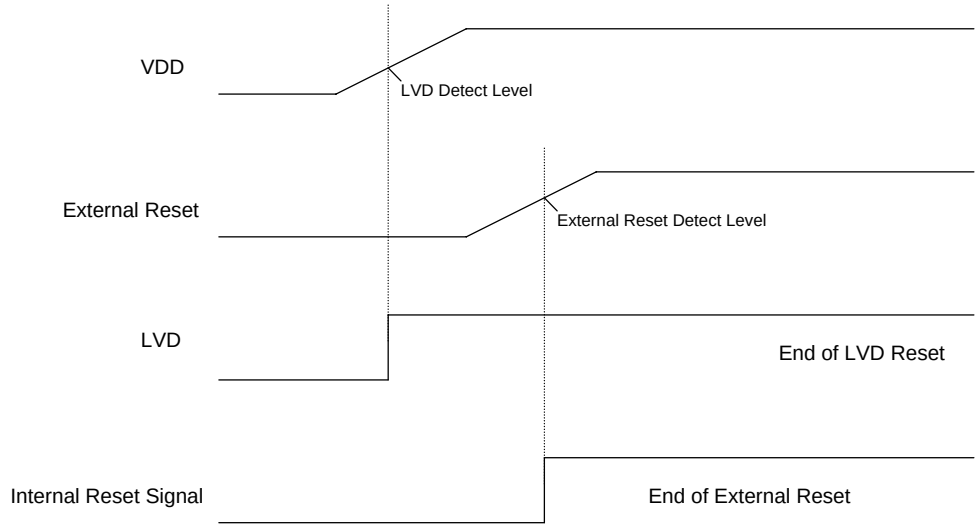
➢ ***SN8PC01***

Stack Level	***STKP Register***			***Stack Buffer***		***Description***
	STKPB2	**STKPB1**	**STKPB0**	**High Byte**	**Low Byte**	
4	0	1	1	STK3H	STK3L	-
3	1	0	0	STK2H	STK2L	-
2	1	0	1	STK1H	STK1L	-
1	1	1	0	STK0H	STK0L	-
0	1	1	1	Free	Free	-

3 RESET

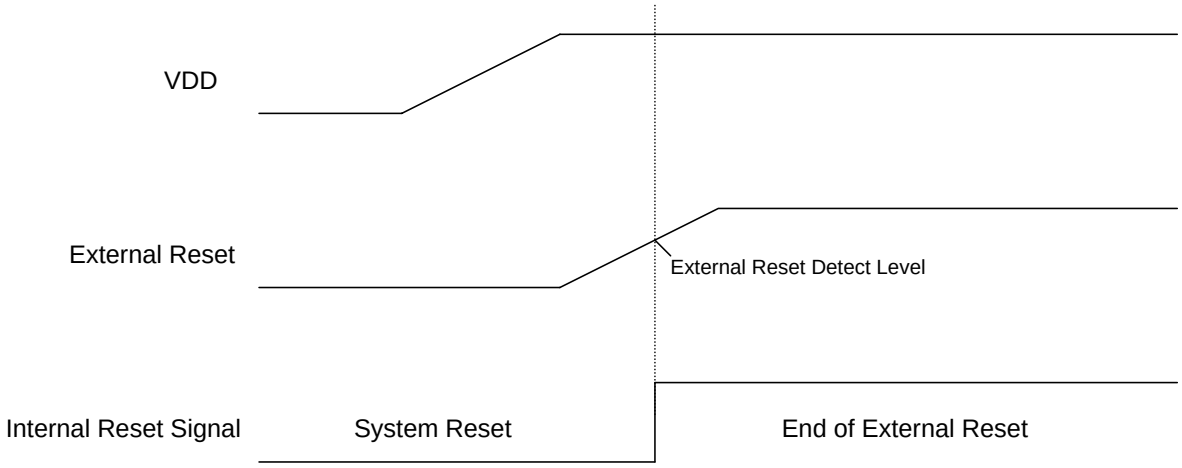
3.1 OVERVIEW

SN8PC01 provides two system resets. One is external reset and the other is internal low voltage detector (LVD). The timing diagram is as the following.



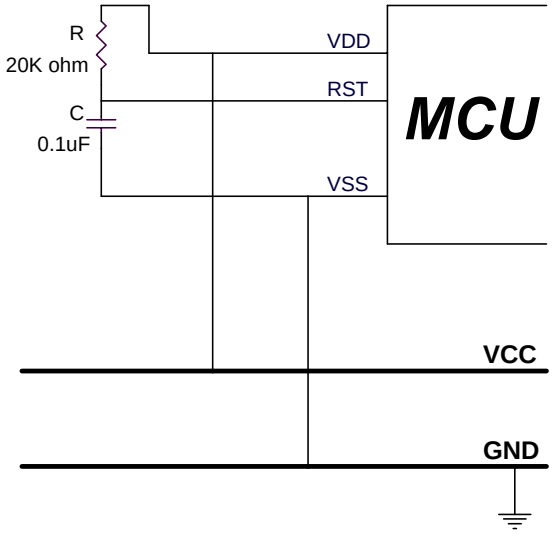
3.2 EXTERNAL RESET DESCRIPTION

The external reset is a low level active device. The reset pin receives the low voltage and resets the system. When the voltage detects high level, it stops resetting the system. Users can use an external reset circuit to control system operation.

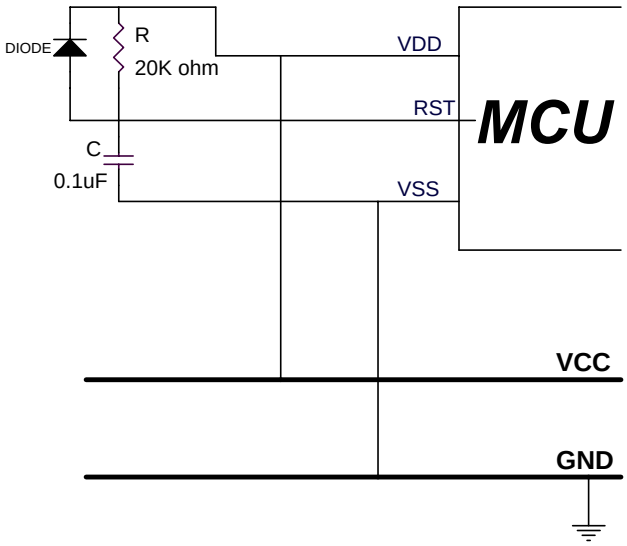


3.2.1 EXTERNAL RESET CIRCUIT

Users must make sure the VDD is stable earlier than external reset. Otherwise, the power on reset maybe fail. The external reset circuit is a simple RC circuit as the following figure.

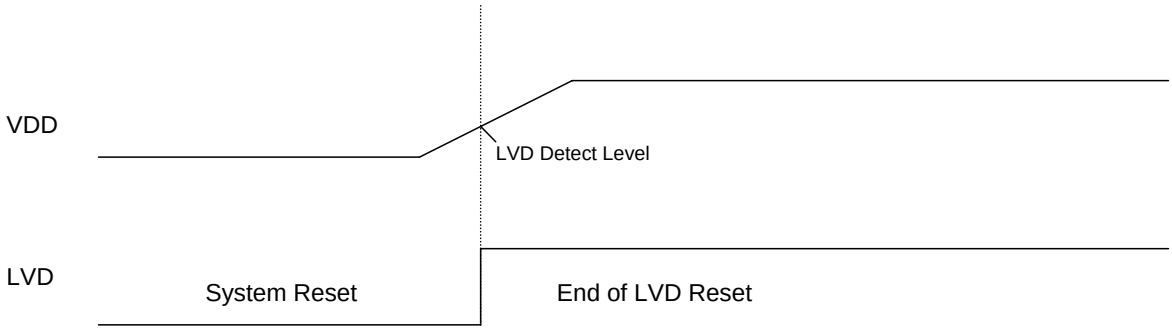


Under different environment, by placing a diode in between VCC and reset pin will help the Brownout reset.



3.3 LOW VOLTAGE DETECTOR (LVD)

The LVD is a low voltage detector. It detects VDD level and reset the system as the VDD lower than the detected voltage. The detect level is 1.8V. If the VDD lower than 1.8V, the system resets.



4 OSCILLATOR AND SYSTEM CLOCK

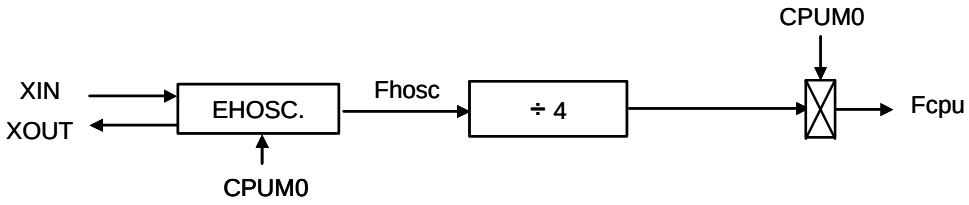
4.1 OVERVIEW

The SN8PC01 is a signal clock micro-controller system. There is high-speed clock. The high-speed clock is generated from the external 455KHz crystal.

The high-speed clock can be system clock (Fosc). The system clock is divided by 4 to be the instruction cycle (Fcpu).

Fcpu = Fosc/4

4.2 CLOCK BLOCK DIAGRAM



➢ EHOSC: External high-speed clock

4.3 OSCM REGISTER DESCRIPTION

The OSCM register is an oscillator control register. It controls oscillator status, system mode.

0CAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCM	0	WDRST	-	-	CPUM0	-	-	-
Read/Write	-	R/W	-	-	R/W	-	-	-
After reset	-	0	-	-	0	-	-	-

WDRST: Reset watchdog timer..

CPUM0: CPU operating mode control bit.
0 = normal
1 = sleep (power down) mode

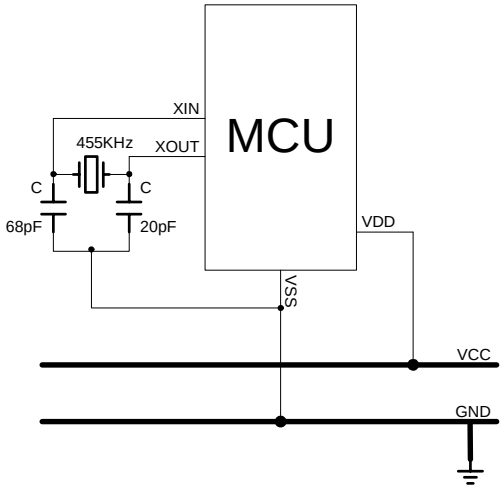
➲ **Example: Reset watchdog timer.**

```
BSET    FWDRST      ; To reset watchdog timer.
```

➲ **Example: When entering the Power Down mode, both high-speed oscillator and internal low-speed oscillator will be stopped.**

```
BSET    FCPUM0      ; To stop external high-speed oscillator and internal low-speed
                    ; oscillator called power down mode (sleep mode).
```

4.4 EXTERNAL SYSTEM OSCILLATOR CIRCUITS



455KHz Crystal

- ***Note: The capacitor connected with XIN must be 68pF. The capacitor connected with XOUT must be 20pF.***
- ***Note: The external oscillator circuit must be directly from Vss pin of micro-controller.***

4.5 SYSTEM OPERATING MODE

The chip is featured with low power consumption by switching around two different modes (Normal mode/Power down mode (Sleep mode)).

- High-speed mode
- Power-down mode (Sleep mode)

Operating mode description

MODE	NORMAL	POWER DOWN (SLEEP)	REMARK
EHOSC	Running	Stop	
CPU instruction	Executing	Stop	
Watchdog timer	Active	Stop	
Wakeup source	-	P0, P1, Reset	

4.5.1 SYSTEM MODE SWITCHING

Switch normal mode to power down (sleep) mode.

```
BSET    FCPUM0    ; Set CPUM0 = 1.
```

- ***During the sleep, only the wakeup pin and reset can wakeup the system back to the normal mode.***

4.5.2 HARDWARE WAKEUP

Under power down mode (sleep mode), P0 and P1 with wakeup function are able to wake the system up. The wakeup function always enables. The hardware wakeup signal is level change.

5 I/O PORT

5.1 I/O PORT MODE

The port direction is fixed by hardware. Port 0 is input direction without pull-up resistors. Port 1 is input direction with pull-up resistors. P2.0~p2.3 are output direction. P2.4, P2.5 are bi-direction I/O. P5.4 is general purpose output and 38KHz signal output controlled by TC0OUT bit. **After power on reset and external reset, P5.4 is logic high status.**

5.2 I/O I/O PORT DATA REGISTER

0D0H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	-	-	-	-	-	-	-	P00
Read/Write	-	-	-	-	-	-	-	R
After reset	-	-	-	-	-	-	-	0

0D1H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

0D2H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2	-	-	P25	P24	P23	P22	P21	P20
Read/Write	-	-	R/W	R/W	W	W	W	W
After reset	-	-	0	0	0	0	0	0

0D5H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P5	-	-	-	P54	-	-	-	-
Read/Write	-	-	-	W	-	-	-	-
After reset	-	-	-	0	-	-	-	-

➲ **Example: Read data from input port.**

```
MOV     A, P0       ; Read data from Port 0
MOV     A, P1       ; Read data from Port 1
```

➲ **Example: Write data to output port.**

```
MOV     A, #0FFH    ; Write data FFH to all Port.
MOV     P2, A
MOV     P5, A
```

➲ **Example: Write one bit data to output port.**

```
MOV     A, #1       ; Set P2.0 to be high.
OR      A, P2
MOV     P2, A
```

➲ **Example: Port bit test.**

```
BTS1    P0.0        ; Bit test 1 for P0.0
BTS0    P1.2        ; Bit test 0 for P1.2
```

5.3 PORT MODE REGISTER

0C2H	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2M	-	-	P25M	P24M	-	-	-	-
Read/Write	-	-	W	W	-	-	-	-
After reset	-	-	0	0	-	-	-	-

➲ **Example: Write data to P2.4, P2.5.**

```
MOV     A, #0FFH        ; Set P2.4, P2.5 as output mode.
MOV     P2M, A
MOV     A, #00010000H   ; Set P2.4 = 1, P2.5 = 0.
MOV     P2, A
```

6 TIMERS

6.1 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, watchdog timer overflow signal raises and resets MCU. Watchdog clock controlled by code option and the clock source is internal low-speed oscillator (ILOSC, 16K @3V, 32K @5V).

- ***Watchdog overflow time is about 0.5 sec @3V, 0.25 sec @5V.***
- ***If watchdog enable, it stops running under power down mode.***

Watchdog clear is controlled by WDRST bit. WDRST =1 is to reset watchdog timer.

0CAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCM	0	WDRST	-	-	CPUM0	-	-	-
Read/Write	-	R/W	-	-	R/W	-	-	-
After reset	-	0	-	-	0	-	-	-

➲ **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.**

```
Main:
        BSET    FWDRST      ; Clear the watchdog timer.
        .       .
        CALL    SUB1
        CALL    SUB2
        .       .
        .       .
        .       .
        JMP     MAIN
```

6.2 38KHz IR OUTPUT

SN8PC01 is build-in 38KHz signal output for IR. The 38KHz output is controlled by TC0OUT bit of TC0M register. The P54 bit of P5 data buffer must be “Low” and the 38KHz signal outputs from P5.4 pins. If P54 bit of P5 data buffer is “High”, P5.4 keeps logic high status. **After power on reset and external reset, P5.4 is logic high status.**

0DAH	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TC0M	-	-	-	-	-	-	TC0OUT	-
Read/Write	-	-	-	-	-	-	R/W	-
After reset	-	-	-	-	-	-	0	-

TC0OUT: 38KHz toggle signal output control bit. **Only valid when P5.4 outputs low.**

0 = Disable, P5.4 is I/O function.
1 = Enable, P54=0 and P5.4 outputs 38KHz signal. P54=1 and P5.4 keeps high.

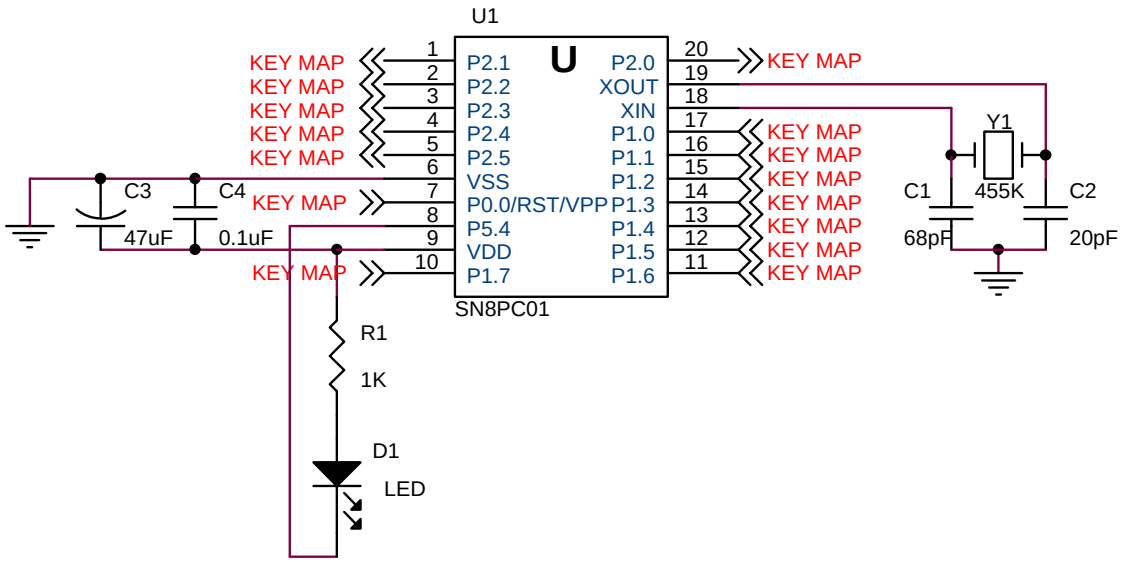
➲ **Example: Setup 38KHz signal output.**

```
BCLR    P5.4        ; Set P5.4 is logic low status.
BSET    FTC0OUT     ; Enable 38KHz signal output to P5.4.
```

6.2.1 IR APPLICATION CIRCUIT

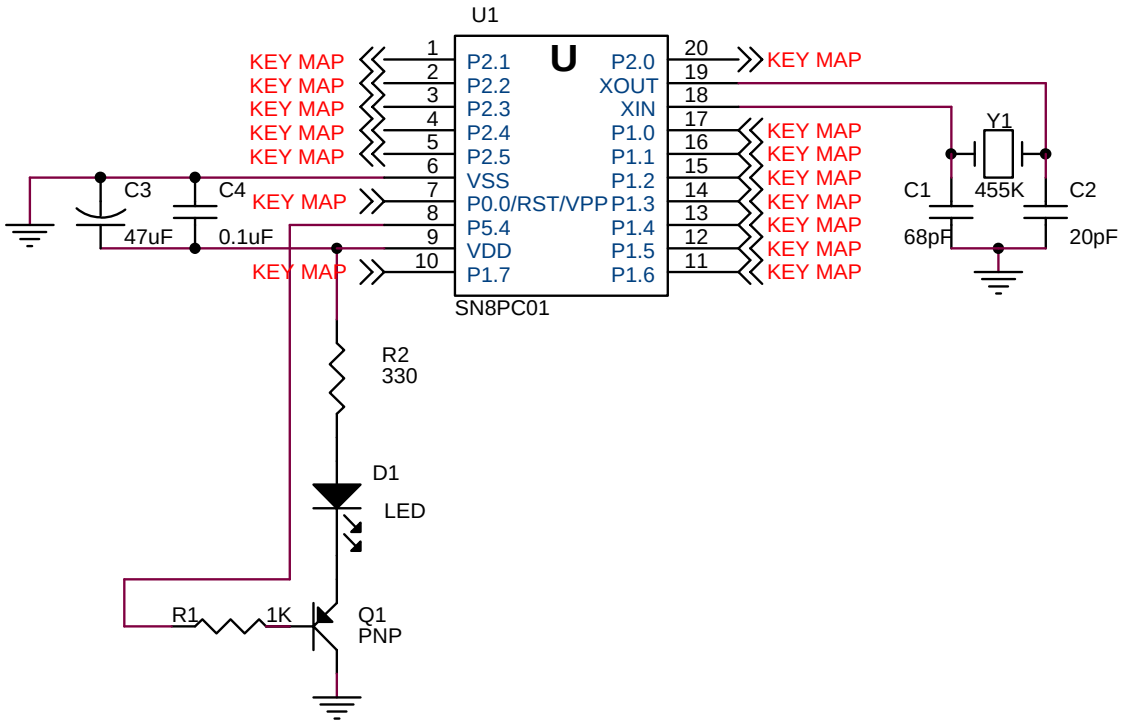
Short Distance:

Sink IR current into P5.4. The current is about 20mA.



Long Distance:

Use a PNP transistor to drive IR LED. The driver current is larger than P5.4 sink current.



7 CODING ISSUE

7.1 TEMPLATE CODE

```
;************************************************************************
; FILENAME  : TEMPLATE.ASM
; AUTHOR    : SONiX
; PURPOSE   : Template Code for SN8PC01
; REVISION  : 04/12/2003 V1.0   First issue
;************************************************************************
;*  (c) Copyright 2003, SONiX TECHNOLOGY CO., LTD.
;************************************************************************

CHIP    SN8PC01                 ; Select the CHIP

;------------------------------------------------------------------------
;                           Include Files
;------------------------------------------------------------------------
.nolist                             ; do not list the macro file

    INCLUDESTD      MACRO1.H
    INCLUDESTD      MACRO2.H
    INCLUDESTD      MACRO3.H

.list                               ; Enable the listing function


;------------------------------------------------------------------------
;                           Constants Definition
;------------------------------------------------------------------------
;   ONE         EQU     1

;------------------------------------------------------------------------
;                           Variables Definition
;------------------------------------------------------------------------
.DATA
                org     0h          ;Data section start from RAM address 0
    Wk00        DS      1           ;Temporary buffer for main loop

;------------------------------------------------------------------------
;                           Bit Variables Definition
;------------------------------------------------------------------------

    Wk00B0      EQU    Wk00.0       ;Bit 0 of Wk00

;------------------------------------------------------------------------
;                           Code section
;------------------------------------------------------------------------

.CODE

    ORG         0                   ;Code section start
;------------------------------------------------------------------------
```

```
; Program reset section
;--------------------------------------------------------------------------
Reset:
    mov         A,#07Fh             ;Initial stack pointer and
    b0mov       STKP,A              ;disable global interrupt
    b0mov       PFLAG,#00h          ;pflag = x,x,x,x,x,c,dc,z
    mov         A,#40h              ;Clear watchdog timer and initial system mode
    b0mov       OSCM,A

    call        ClrRAM              ;Clear RAM
    call        SysInit             ;System initial

;--------------------------------------------------------------------------
; Main routine
;--------------------------------------------------------------------------
Main:
    b0bset      FWDRST              ;Clear watchdog timer

    call        MnApp

    jmp         Main

;--------------------------------------------------------------------------
; Main application
;--------------------------------------------------------------------------
MnApp:

    ; Put your main program here

    ret


;--------------------------------------------------------------------------
; SysInit
; System initial to define Register, RAM, I/O, Timer......
;--------------------------------------------------------------------------
SysInit:

    ret

;--------------------------------------------------------------------------
; ClrRAM
; Use index @YZ to clear RAM (00h~1Fh)
;--------------------------------------------------------------------------

ClrRAM:

    clr         Y                   ;
    mov         Z,#0x1f             ;Set @YZ address from 1fh

ClrRAM10:
    clr         @YZ                 ;Clear @YZ content
    decms       Z                   ;z = z – 1 , skip next if z=0
    jmp         ClrRAM10
    clr         @YZ                 ;Clear address $00

    ret

;--------------------------------------------------------------------------
    ENDP
```

7.2 PROGRAM CHECK LIST

Item	Description
Undefined Bits	All bits those are marked as “0” (undefined bits) in system registers should be set “0” to avoid unpredicted system errors.
Non-Used I/O	Non-used I/O ports should be set as output low mode or pull-up at input mode to save current consumption.
Sleep Mode	Enable on-chip pull-up resisters of port 0 and port 1 to avoid unpredicted wakeup.
Stack Buffer	Be careful of function call and interrupt service routine operation. Don’t let stack buffer overflow or underflow.
System Initial	1. Write 0x7F into STKP register to initial stack pointer and disable global interrupt 2. Clear all RAM. 3. Initialize all system register even unused registers.

8 INSTRUCTION SET TABLE

Field	Mnemonic		Description	C	Z	Cycle
MOVE	MOV	A, M	A ← M	-	√	1
	MOV	M, A	M ← A	-	-	1
	MOV	A, I	A ← I	-	-	1
	MOVC		R, A ← ROM [Y,Z]	-	-	2
ARITHMETICM	ADC	A, M	A ← A + M + C, if occur carry, then C=1, else C=0	√	√	1
	SBC	A, M	A ← A – M - /C, if occur borrow, then C=0, else C=1	√	√	1
LOGIC	AND	A, M	A ← A and M	-	√	1
	OR	A, M	A ← A or M	-	√	1
	XOR	A, M	A ← A xor M	-	√	1
PROCESS	RRC	M	A ← RRC M	√	-	1
	BCLR	M. b	M.b ← 0	-	-	1
	BSET	M. b	M.b ← 1	-	-	1
BRANCH	CMPRS	A, I	ZF,C ← A – I, If A = I, then skip next instruction	√	√	1 + S
	INCMS	M	M ← M + 1, If M = 0, then skip next instruction	-	-	1 + S
	DECMS	M	M ← M – 1, If M = 0, then skip next instruction	-	-	1 + S
	BTS0	M. b	If M.b = 0, then skip next instruction	-	-	1 + S
	BTS1	M. b	If M.b = 1, then skip next instruction	-	-	1 + S
	JMP	d	PC15/14 ← RomPages1/0, PC13~PC0 ← d	-	-	2
	CALL	d	Stack ← PC15~PC0, PC15/14 ← RomPages1/0, PC13~PC0 ← d	-	-	2
MIS	RET		PC ← Stack	-	-	2
	NOP		No operation	-	-	1

Note : a). Working registers = R, Y and Z
b). The memory is access to location RAM[Y,Z], if M = @YZ (located at address E7H in RAM bank 0).
c). All instructions are one cycle except for program branch and PC update which are two cycles.

9 ELECTRICAL CHARACTERISTIC

9.1 Absolute Maximum Rating

Supply voltage (Vdd) ..... - 0.3V ~ 6.0V
Input in voltage (Vin) ..... Vss – 0.2V ~ Vdd + 0.2V
Operating ambient temperature (Topr) ..... 0°C ~ + 70°C
Storage ambient temperature (Tstor) ..... –30°C ~ + 125°C
Power consumption (Pc) 500mW

9.2 Electrical Characteristic

(All of voltages refer to Vss, Vdd = 5.0V, fosc = 455KHz, ambient temperature is 25℃ unless otherwise note.)

PARAMETER	*SYM.*	*DESCRIPTION*		*MIN.*	*TYP.*	*MAX.*	*UNIT*
Operating voltage	Vdd	Normal mode, Vpp = Vdd		2.2	3.0	5.5	V
		Programming mode, Vpp = 12.3V		4.5	5.0	5.5	
OTP programming voltage	Vpp	OTP programming voltage		-	12.3	-	V
RAM Data Retention voltage	Vdr			-	1.5	-	V
Internal POR	Vpor	Vdd rise rate to ensure internal power-on reset		-	0.05	-	V/ms
Input Low Voltage	ViL1	Input with Schmitt trigger buffer		Vss	-	0.3Vdd	V
	ViL2	Reset pin ; Xin (in RC mode)		Vss	-	0.3Vdd	V
	ViL3	Xin (in X'tal mode)		Vss	-	0.3Vdd	V
Input High Voltage	ViH1	Input with Schmitt trigger buffer		0.7Vdd	-	Vdd	V
	ViH2	Reset pin ; Xin (in RC mode)		0.7Vdd	-	Vdd	V
	ViH3	Xin (in X'tal mode)		0.7Vdd	-	Vdd	V
Reset pin leakage current	Ilekg	Vin = Vdd		-	-	1	uA
I/O port pull-up resistor	Rup	Vin = Vss , Vdd = 5V		-	100	-	KΩ
I/O port input leakage current	Ilekg	Pull-up resistor disable, Vin = Vdd		-	-	2	uA
Port 2 source current	IoH	Vop = Vdd – 0.5V		-	4	-	mA
sink current	IoL	Vop = Vss + 0.5V		-	4	-	
Port 5 source current	IoH	Vop = Vdd – 0.5V		-	12	-	mA
sink current	IoL	Vop = Vss + 0.5V		-	20	-	
Supply Current	Idd1	Run Mode, no loading	Vdd= 3V 455Khz	-	1	-	mA
	Idd2	Sleep Mode	Vdd= 3V	-	1		uA
LVD detect level	V_{LVD}	Internal POR detect level		-	1.8	-	V

10 DEVELOPMENT TOOL

SN8PC01 development tools are as following.

- **ICE version: S8KD-2 ICE.**
- **IDE version: SN8C01IDE_V101.**
- **Writer: Writer V3.0, EZ-writer, MP-writer.**

In emulation duration, the crystal setting of S8KD-2 ICE is 455KHz crystal. The High_CLK selection of ICE SW1 switch is “X’tal 4M” (S3, S2=11). Y1 is 455KHz crystal and C3, C4 are 100pF. In OTP program using EZ-writer, the crystal must be change to 4MHz and C3, C4 are 20pF.

11 TRANSITION SOCKET

GND	1	2	VDD
CE	3	4	CLK
OE	5	6	PGM
D0	7	8	D1
D2	9	10	D3
D4	11	12	D5
D6	13	14	D7
VPP	15	16	VDD
RST	17	18	HLS

SONiX Writer V2.5 JP1 Pin Allocation

GND	2	1	VDD
CE	4	3	CLK
OE	6	5	PGM
D0	8	7	D1
D2	10	9	D3
D4	12	11	D5
D6	14	13	D7
VPP	16	15	VDD
RST	18	17	HLS
	20	19	

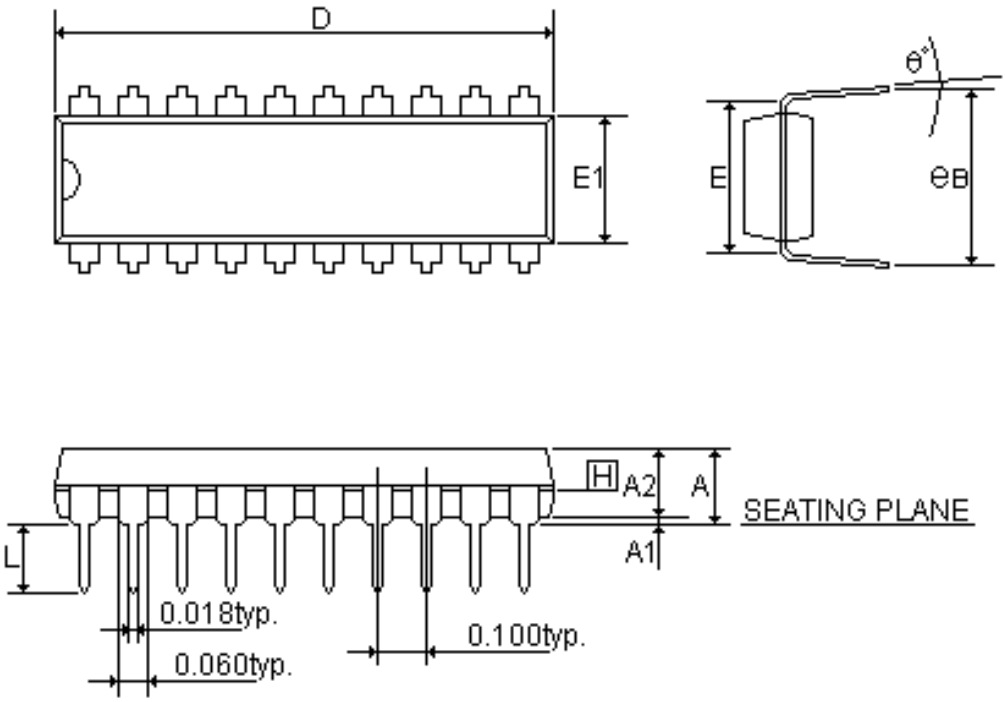
SONiX Writer V3.0 JP1 Pin Allocation

This Serial Connector locates at lower right hand side of the writer board. Users are welcome to design their own transition socket module. However, please be sure to read this chapter and check the pins of the OTP devices. Following chart provides the references for SONiX Writer and Transition Socket.

Writer V2.5 Connector		Writer V3.0 Connector		OTP Name							
				SN8PC01							
number	name	number	Name	Number	Pin						
2	VDD	1	VDD	9	VDD						
1	GND	2	GND	6	VSS						
4	CLK	3	CLK	18	XIN						
3	CE	4	CE	13	P1.4						
6	PGM	5	PGM	11	P1.6						
5	OE	6	OE	12	P1.5						
8	D1	7	D1	16	P1.1						
7	D0	8	D0	17	P1.0						
10	D3	9	D3	14	P1.3						
9	D2	10	D2	15	P1.2						
12	D5	11	D5	1	P2.1						
11	D4	12	D4	20	P2.0						
14	D7	13	D7	3	P2.3						
13	D6	14	D6	2	P2.2						
16	VDD	15	VDD		-						
15	VPP	16	VPP	7	RST						
18	HLS	17	HLS	10	P1.7						
17	RST	18	RST	-	-						
-	-	19~20	-	-	-						

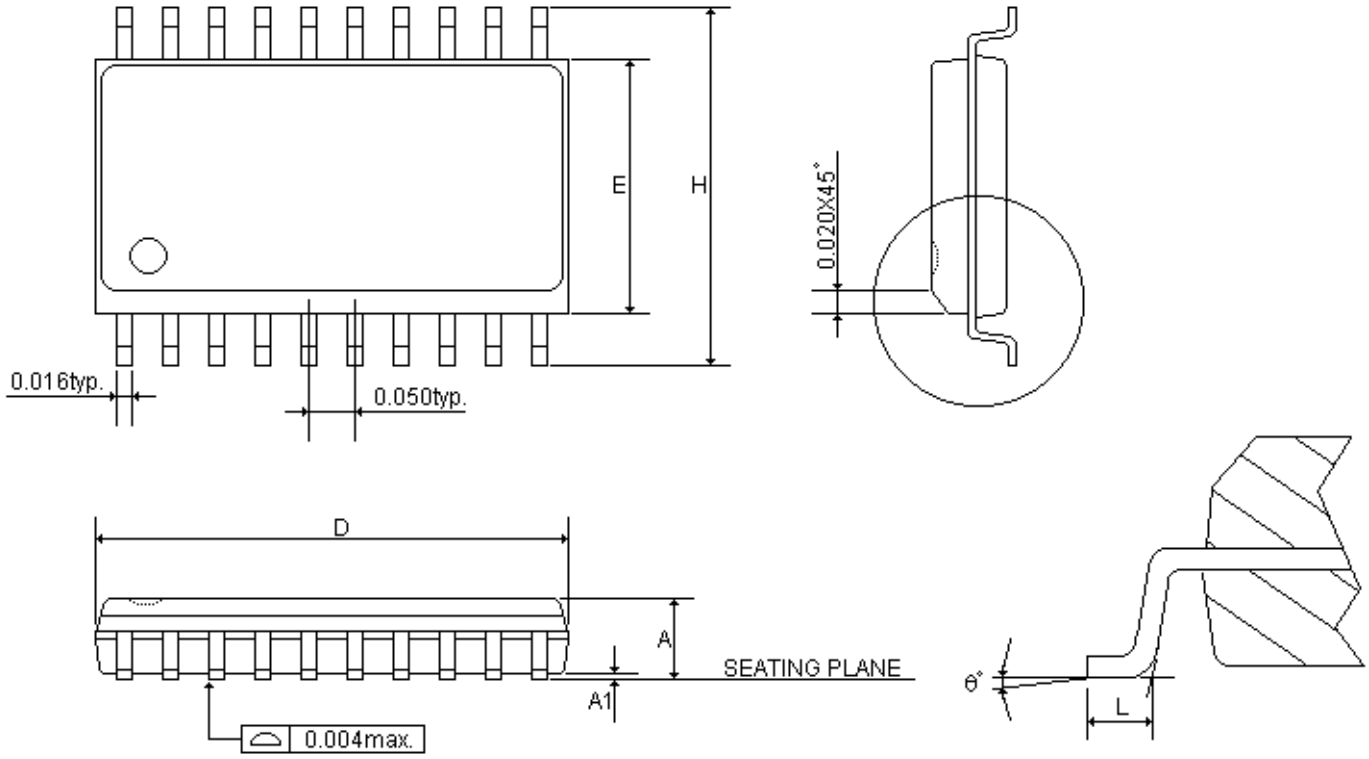
12 PACKAGE INFORMATION

12.1 P-DIP 20 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	-	-	0.210	-	-	5.334
A1	0.015	-	-	0.381	-	-
A2	0.125	0.130	0.135	3.175	3.302	3.429
D	0.98	1.030	1.060	24.892	26.162	26.924
E		0.300			7.62	
E1	0.245	0.250	0.255	6.223	6.35	6.477
L	0.115	0.130	0.150	2.921	3.302	3.810
eB	0.335	0.355	0.375	8.509	9.017	9.525
θ°	0°	7°	15°	0°	7°	15°

12.2 SOP 20 PIN



SYMBOLS	MIN	NOR	MAX	MIN	NOR	MAX
	(inch)			(mm)		
A	0.093	0.099	0.104	2.362	2.502	2.642
A1	0.004	0.008	0.012	0.102	0.203	0.305
D	0.496	0.502	0.508	12.598	12.751	12.903
E	0.291	0.295	0.299	7.391	7.493	7.595
H	0.394	0.407	0.419	10.008	10.325	10.643
L	0.016	0.033	0.050	0.406	0.838	1.270
θ°	0°	4°	8°	0°	4°	8°

Main Office:
Address: 9F, NO. 8, Hsien Cheng 5th St, Chupei City, Hsinchu, Taiwan R.O.C.
Tel: 886-3-551 0520
Fax: 886-3-551 0523
Taipei Office:
Address: 15F-2, NO. 171, Song Ted Road, Taipei, Taiwan R.O.C.
Tel: 886-2-2759 1980
Fax: 886-2-2759 8180
Hong Kong Office:
Address: Flat 3 9/F Energy Plaza 92 Granville Road, Tsimshatsui East Kowloon.
Tel: 852-2723 8086
Fax: 852-2723 9179
Technical Support by Email:
Sn8fae@sonix.com.tw