Nuvoton Technology Corporation - N76E003AT20 Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, I²S, POR, PWM, WDT
Number of I/O	18
Program Memory Size	18KB (18K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.4V ~ 5.5V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/nuvoton-technology-corporation-america/n76e003at20

Bit	Name	Description
3	PT1H	**Timer 1 interrupt priority high bit**
2	PX1H	**External interrupt 1 priority high bit**
1	PT0H	**Timer 0 interrupt priority high bit**
0	PX0H	**External interrupt 0 priority high bit**

[2] IPH is used in combination with the IP respectively to determine the priority of each interrupt source. See Table 20-2. Interrupt Priority Level Setting for correct interrupt priority configuration.

PWMINTC – PWM Interrupt Control

7	6	5	4	3	2	1	0
-	-	INTTYP1	INTTYP0	-	INTSEL2	INTSEL1	INTSEL0
-	-	R/W	R/W	-	R/W	R/W	R/W

Address: B7H, Page:1 Reset value: 0000 0000b

Bit	Name	Description
5:4	INTTYP[1:0]	**PWM interrupt type select** These bit select PWM interrupt type. 00 = Falling edge on PWM0/1/2/3/4/5 pin. 01 = Rising edge on PWM0/1/2/3/4/5 pin. 10 = Central point of a PWM period. 11 = End point of a PWM period. Note that the central point interrupt or the end point interrupt is only available while PWM operates in center-aligned type.
2:0	INTSEL[2:0]	**PWM interrupt pair select** These bits select which PWM channel asserts PWM interrupt when PWM interrupt type is selected as falling or rising edge on PWM0/1/2/3/4/5 pin.. 000 = PWM0. 001 = PWM1. 010 = PWM2. 011 = PWM3. 100 = PWM4. 101 = PWM5. Others = PWM0.

IP – Interrupt Priority (Bit-addressable)[1]

7	6	5	4	3	2	1	0
-	PADC	PBOD	PS	PT1	PX1	PT0	PX0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: B8H Reset value: 0000 0000b

Bit	Name	Description
6	PADC	**ADC interrupt priority low bit**
5	PBOD	**Brown-out detection interrupt priority low bit**
4	PS	**Serial port 0 interrupt priority low bit**
3	PT1	**Timer 1 interrupt priority low bit**
2	PX1	**External interrupt 1 priority low bit**

I2CON – I²C Control (Bit-addressable)

7	6	5	4	3	2	1	0
-	I2CEN	STA	STO	SI	AA	-	I2CPX
-	R/W	R/W	R/W	R/W	R/W	-	R/W

Address: C0H Reset value: 0000 0000b

Bit	Name	Description
6	I2CEN	**I²C bus enable** 0 = I²C bus Disabled. 1 = I²C bus Enabled. Before enabling the I²C, SCL and SDA port latches should be set to logic 1.
5	STA	**START flag** When STA is set, the I²C generates a START condition if the bus is free. If the bus is busy, the I²C waits for a STOP condition and generates a START condition following. If STA is set while the I²C is already in the master mode and one or more bytes have been transmitted or received, the I²C generates a repeated START condition. Note that STA can be set anytime even in a slave mode, but STA is not hardware automatically cleared after START or repeated START condition has been detected. User should take care of it by clearing STA manually.
4	STO	**STOP flag** When STO is set if the I²C is in the master mode, a STOP condition is transmitted to the bus. STO is automatically cleared by hardware once the STOP condition has been detected on the bus. The STO flag setting is also used to recover the I²C device from the bus error state (I2STAT as 00H). In this case, no STOP condition is transmitted to the I²C bus. If the STA and STO bits are both set and the device is original in the master mode, the I²C bus will generate a STOP condition and immediately follow a START condition. If the device is in slave mode, STA and STO simultaneous setting should be avoid from issuing illegal I²C frames.
3	SI	**I²C interrupt flag** SI flag is set by hardware when one of 26 possible I²C status (besides F8H status) is entered. After SI is set, the software should read I2STAT register to determine which step has been passed and take actions for next step. SI is cleared by software. Before the SI is cleared, the low period of SCL line is stretched. The transaction is suspended. It is useful for the slave device to deal with previous data bytes until ready for receiving the next byte. The serial transaction is suspended until SI is cleared by software. After SI is cleared, I²C bus will continue to generate START or repeated START condition, STOP condition, 8-bit data, or so on depending on the software configuration of controlling byte or bits. Therefore, user should take care of it by preparing suitable setting of registers before SI is software cleared.

PWMPH – PWM Period High Byte

7	6	5	4	3	2	1	0
PWMP[15:8]							
R/W							

Address: D1H reset value: 0000 0000b

Bit	Name	Description
7:0	PWMP[15:8]	**PWM period high byte** This byte with PWMPL controls the period of the PWM generator signal.

PWM0H – PWM0 Duty High Byte

7	6	5	4	3	2	1	0
PWM0[15:8]							
R/W							

Address: D2H reset value: 0000 0000b

Bit	Name	Description
7:0	PWM0[15:8]	**PWM0 duty high byte** This byte with PWM0L controls the duty of the output signal PG0 from PWM generator.

PWM1H – PWM1 Duty High Byte

7	6	5	4	3	2	1	0
PWM1[15:8]							
R/W							

Address: D3H reset value: 0000 0000b

Bit	Name	Description
7:0	PWM1[15:8]	**PWM1 duty high byte** This byte with PWM1L controls the duty of the output signal PG1 from PWM generator.

PWM2H – PWM2 Duty High Byte

7	6	5	4	3	2	1	0
PWM2[15:8]							
R/W							

Address: D4H reset value: 0000 0000b

Bit	Name	Description
7:0	PWM2[15:8]	**PWM2 duty high byte** This byte with PWM2L controls the duty of the output signal PG2 from PWM generator.

PWMCON0 – PWM Control 0 (Bit-addressable)

7	6	5	4	3	2	1	0
PWMRUN	LOAD	PWMF	CLRPWM	-	-	-	-
R/W	R/W	R/W	R/W	-	-	-	-

Address: D8H Reset value: 0000 0000b

Bit	Name	Description
7	PWMRUN	**PWM run enable** 0 = PWM stays in idle. 1 = PWM starts running.
6	LOAD	**PWM new period and duty load** This bit is used to load period and duty control registers in their buffer if new period or duty value needs to be updated. The loading will act while a PWM period is completed. The new period and duty affected on the next PWM cycle. After the loading is complete, LOAD will be automatically cleared via hardware. The meaning of writing and reading LOAD bit is different. Writing: 0 = No effect. 1 = Load new period and duty in their buffers while a PWM period is completed. Reading: 0 = A loading of new period and duty is finished. 1 = A loading of new period and duty is not yet finished.
5	PWMF	**PWM flag** This flag is set according to definitions of INTSEL[2:0] and INTTYP[1:0] in PWMINTC. This bit is cleared by software.
4	CLRPWM	**Clear PWM counter** Setting this bit clears the value of PWM 16-bit counter for resetting to 0000H. After the counter value is cleared, CLRPWM will be automatically cleared via hardware. The meaning of writing and reading CLRPWM bit is different. Writing: 0 = No effect. 1 = Clearing PWM 16-bit counter. Reading: 0 = PWM 16-bit counter is completely cleared. 1 = PWM 16-bit counter is not yet cleared.

PWMPL – PWM Period Low Byte

7	6	5	4	3	2	1	0
PWMP[7:0]							
R/W							

Address: D9H reset value: 0000 0000b

Bit	Name	Description
7:0	PWMP[7:0]	**PWM period low byte** This byte with PWMPH controls the period of the PWM generator signal.

ADCCON0 – ADC Control 0 (Bit-addressable)

7	6	5	4	3	2	1	0
ADCF	ADCS	ETGSEL1	ETGSEL0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: E8H Reset value: 0000 0000b

Bit	Name	Description
7	ADCF	**ADC flag** This flag is set when an A/D conversion is completed. The ADC result can be read. While this flag is 1, ADC cannot start a new converting. This bit is cleared by software.
6	ADCS	**A/D converting software start trigger** Setting this bit 1 triggers an A/D conversion. This bit remains logic 1 during A/D converting time and is automatically cleared via hardware right after conversion complete. The meaning of writing and reading ADCS bit is different. Writing: 0 = No effect. 1 = Start an A/D converting. Reading: 0 = ADC is in idle state. 1 = ADC is busy in converting.
5:4	ETGSEL[1:0]	**External trigger source select** When ADCEX (ADCCON1.1) is set, these bits select which pin output triggers ADC conversion. 00 = PWM0. 01 = PWM2. 10 = PWM4. 11 = STADC pin.
3:0	ADCHS[3:0]	**A/D converting channel select** This filed selects the activating analog input source of ADC. If ADCEN is 0, all inputs are disconnected. 0000 = AIN0. 0001 = AIN1. 0010 = AIN2. 0011 = AIN3. 0100 = AIN4. 0101 = AIN5. 0110 = AIN6. 0111 = AIN7 1000 = Internal band-gap voltage. Others = Reserved.

capture, or any edge capture. Each input capture channel has to set its own enabling bit CAPEN0~2 (CAPCON0[6:4]) before use.

While input capture channel is enabled and the selected edge trigger occurs, the content of the free running Timer 2 counter, TH2 and TL2, will be captured, transferred, and stored into the capture registers CnH and CnL. The edge triggering also causes CAPFn (CAPCON0.n) set by hardware. The interrupt will also generate if the ECAP (EIE.2) and EA bit are both set. For three input capture flags share the same interrupt vector, user should check CAPFn to confirm which channel comes the input capture edge. These flags should be cleared by software.

The bit CAPCR (CAPCON2.3) benefits the implement of period calculation. Setting CAPCR makes the hardware clear Timer 2 as 0000H automatically after the value of TH2 and TL2 have been captured after an input capture edge event occurs. It eliminates the routine software overhead of writing 16-bit counter or an arithmetic subtraction.

CAPCON0 – Input Capture Control 0

7	6	5	4	3	2	1	0
-	CAPEN2	CAPEN1	CAPEN0	-	CAPF2	CAPF1	CAPF0
-	R/W	R/W	R/W	-	R/W	R/W	R/W

Address: 92H Reset value: 0000 0000b

Bit	Name	Description
6	CAPEN2	**Input capture 2 enable** 0 = Input capture channel 2 Disabled. 1 = Input capture channel 2 Enabled.
5	CAPEN1	**Input capture 1 enable** 0 = Input capture channel 1 Disabled. 1 = Input capture channel 1 Enabled.
4	CAPEN0	**Input capture 0 enable** 0 = Input capture channel 0 Disabled. 1 = Input capture channel 0 Enabled.
2	CAPF2	**Input capture 2 flag** This bit is set by hardware if the determined edge of input capture 2 occurs. This bit should cleared by software.
1	CAPF1	**Input capture 1 flag** This bit is set by hardware if the determined edge of input capture 1 occurs. This bit should cleared by software.
0	CAPF0	**Input capture 0 flag** This bit is set by hardware if the determined edge of input capture 0 occurs. This bit should cleared by software.

CAPCON1 – Input Capture Control 1

7	6	5	4	3	2	1	0
-	-	CAP2LS[1:0]		CAP1LS[1:0]		CAP0LS[1:0]	
-	-	R/W		R/W		R/W	

Address: 93H Reset value: 0000 0000b

Bit	Name	Description
5:4	CAP2LS[1:0]	**Input capture 2 level select** 00 = Falling edge. 01 = Rising edge. 10 = Either Rising or falling edge. 11 = Reserved.
3:2	CAP1LS[1:0]	**Input capture 1 level select** 00 = Falling edge. 01 = Rising edge. 10 = Either Rising or falling edge. 11 = Reserved.
1:0	CAP0LS[1:0]	**Input capture 0 level select** 00 = Falling edge. 01 = Rising edge. 10 = Either Rising or falling edge. 11 = Reserved.

CAPCON2 – Input Capture Control 2

7	6	5	4	3	2	1	0
-	ENF2	ENF1	ENF0	-	-	-	-
-	R/W	R/W	R/W	-	-	-	-

Address: 94H Reset value: 0000 0000b

Bit	Name	Description
6	ENF2	**Enable noise filer on input capture 2** 0 = Noise filter on input capture channel 2 Disabled. 1 = Noise filter on input capture channel 2 Enabled.
5	ENF1	**Enable noise filer on input capture 1** 0 = Noise filter on input capture channel 1 Disabled. 1 = Noise filter on input capture channel 1 Enabled.
4	ENF0	**Enable noise filer on input capture 0** 0 = Noise filter on input capture channel 0 Disabled. 1 = Noise filter on input capture channel 0 Enabled.

C0L – Capture 0 Low Byte

7	6	5	4	3	2	1	0
C0L[7:0]							
R/W							

Address: E4H Reset value: 0000 0000b

Bit	Name	Description
7:0	C0L[7:0]	**Input capture 0 result low byte** The C0L register is the low byte of the 16-bit result captured by input capture 0.

C0H – Capture 0 High Byte

7	6	5	4	3	2	1	0
C0H[7:0]							
R/W							

Address: E5H Reset value: 0000 0000b

Bit	Name	Description
7:0	C0H[7:0]	**Input capture 0 result high byte** The C0H register is the high byte of the 16-bit result captured by input capture 0.

C1L – Capture 1 Low Byte

7	6	5	4	3	2	1	0
C1L[7:0]							
R/W							

Address: E6H Reset value: 0000 0000b

Bit	Name	Description
7:0	C1L[7:0]	**Input capture 1 result low byte** The C1L register is the low byte of the 16-bit result captured by input capture 1.

C1H – Capture 1 High Byte

7	6	5	4	3	2	1	0
C1H[7:0]							
R/W							

Address: E7H Reset value: 0000 0000b

Bit	Name	Description
7:0	C1H[7:0]	**Input capture 1 result high byte** The C1H register is the high byte of the 16-bit result captured by input capture 1.

11. WATCHDOG TIMER (WDT)

The N76E003 provides one Watchdog Timer (WDT). It can be configured as a time-out reset timer to reset whole device. Once the device runs in an abnormal status or hangs up by outward interference, a WDT reset recover the system. It provides a system monitor, which improves the reliability of the system. Therefore, WDT is especially useful for system that is susceptible to noise, power glitches, or electrostatic discharge. The WDT also can be configured as a general purpose timer, of which the periodic interrupt serves as an event timer or a durational system supervisor in a monitoring system, which is able to operate during Idle or Power-down mode. WDTEN[3:0] (CONFIG4[7:4]) initialize the WDT to operate as a time-out reset timer or a general purpose timer.

CONFIG4

7	6	5	4	3	2	1	0
WDTEN[3:0]				-	-	-	-
R/W				-	-	-	-

Factory default value: 1111 1111b

Bit	Name	Description
7:4	WDTEN[3:0]	**WDT enable** This field configures the WDT behavior after MCU execution. 1111 = WDT is Disabled. WDT can be used as a general purpose timer via software control. 0101 = WDT is Enabled as a time-out reset timer and it stops running during Idle or Power-down mode. Others = WDT is Enabled as a time-out reset timer and it keeps running during Idle or Power-down mode.

The WDT is implemented with a set of divider that divides the low-speed internal oscillator clock nominal 10 kHz. The divider output is selectable and determines the time-out interval. When the time-out interval is fulfilled, it will wake the system up from Idle or Power-down mode and an interrupt event will occur if WDT interrupt is enabled. If WDT is initialized as a time-out reset timer, a system reset will occur after a period of delay if without any software action.

WDCON – Watchdog Timer Control (TA protected)

7	6	5	4	3	2	1	0
WDTR	WDCLR	WDTF	WIDPD	WDTRF[1]	WDPS[2:0][2]		
R/W	R/W	R/W	R/W	R/W	R/W		

Address: AAH Reset value: see Table 6-2. SFR Definitions and Reset Values

Bit	Name	Description
7	WDTR	**WDT run** This bit is valid only when control bits in WDTEN[3:0] (CONFIG4[7:4]) are all 1. At this time, WDT works as a general purpose timer. 0 = WDT Disabled. 1 = WDT Enabled. The WDT counter starts running.

NOTICE: WDT counter has been specially taken care. The hardware automatically clears WDT counter and pre-scalar value after :

(1) Entering into or being woken-up from Idle or Power Down mode
(2) Any resets. It prevents unconscious system reset.

The main application of the WDT with time-out reset enabling is for the system monitor. This is important in real-time control applications. In case of some power glitches or electro-magnetic interference, CPU may begin to execute erroneous codes and operate in an unpredictable state. If this is left unchecked the entire system may crash. Using the WDT during software development requires user to select proper “Feeding Dog” time by clearing the WDT counter. By inserting the instruction of setting WDCLR, it allows the code to run without any WDT reset. However If any erroneous code executes by any interference, the instructions to clear the WDT counter will not be executed at the required instants. Thus the WDT reset will occur to reset the system state from an erroneously executing condition and recover the system.

11.2 General Purpose Timer

There is another application of the WDT, which is used as a simple, long period timer. When the CONFIG bits WDTEN[3:0] (CONFIG4[7:4]) is FH, the WDT is initialized as a general purpose timer. In this mode, WDTR and WIDPD are fully accessed via software.

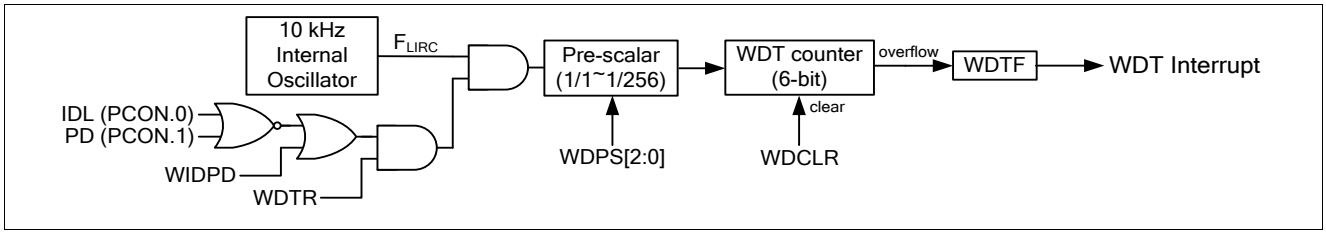

Figure 11.2. Watchdog Timer Block Diagram

The WDT starts running by setting WDTR as 1 and halts by clearing WDTR as 0. The WDTF flag will be set while the WDT completes the selected time interval. The software polls the WDTF flag to detect a time-out. An interrupt will occur if the individual interrupt EWDT (EIE.4) and global interrupt enable EA is set. WDT will continue counting. User should clear WDTF and wait for the next overflow by polling WDTF flag or waiting for the interrupt occurrence.

In some application of low power consumption, the CPU usually stays in Idle mode when nothing needs to be served to save power consumption. After a while the CPU will be woken up to check if anything needs to be served at an interval of programmed period implemented by Timer 0~3. However, the current consumption of Idle mode still keeps at a “mA” level. To further reducing the current consumption to “mA” level, the CPU should stay in Power-down mode when nothing needs to be served, and has the ability of waking up at a

Table 13-2. Serial Port 1 Mode Description

Mode	SM0	SM1	Description	Frame Bits	Baud Rate
0	0	0	Synchronous	8	F_{SYS} divided by 12 or by 2[1]
1	0	1	Asynchronous	10	Timer 3 overflow rate divided by 16
2	1	0	Asynchronous	11	F_{SYS} divided by 32 or 64[2]
3	1	1	Asynchronous	11	Timer 3 overflow rate divided by 16

[1] While SM2_1 (SCON_1.5) is logic 1.
[2] While SMOD_1 (T3CON.7) is logic 1.

SBUF – Serial Port 0 Data Buffer

7	6	5	4	3	2	1	0
SBUF[7:0]							
R/W							

Address: 99H Reset value: 0000 0000b

Bit	Name	Description
7:0	SBUF[7:0]	**Serial port 0 data buffer** This byte actually consists two separate registers. One is the receiving resister, and the other is the transmitting buffer. When data is moved to SBUF, it goes to the transmitting buffer and is shifted for serial transmission. When data is moved from SBUF, it comes from the receiving register. The transmission is initiated through giving data to SBUF.

SBUF_1 – Serial Port 1 Data Buffer

7	6	5	4	3	2	1	0
SBUF_1[7:0]							
R/W							

Address: 9AH Reset value: 0000 0000b

Bit	Name	Description
7:0	SBUF_1[7:0]	**Serial port 1 data buffer** This byte actually consists two separate registers. One is the receiving resister, and the other is the transmitting buffer. When data is moved to SBUF_1, it goes to the transmitting buffer and is shifted for serial transmission. When data is moved from SBUF_1, it comes from the receiving register. The transmission is initiated through giving data to SBUF_1.

Table 13-3. UART Baud Rate Formulas

UART Mode	Baud Rate Clock Source	Baud Rate	Formula Number
0	System clock	$F_{SYS}/12$ or $F_{SYS}/2$ [1]	1
2	System clock	$F_{SYS}/64$ or $F_{SYS}/32$ [2]	2
1 or 3	Timer 1 (only for UART0)[3]	$\frac{2^{SMOD}}{32}\times\frac{F_{SYS}}{12\times(256-TH1)}$ or $\frac{2^{SMOD}}{32}\times\frac{F_{SYS}}{256-TH1}$ [4]	3
	Timer 3 (for UART0)	$\frac{2^{SMOD}}{32}\times\frac{F_{SYS}}{Pre-scale\times(65536-\{RH3,RL3\})}$ [5]	4
	Timer 3 (for UART1)	$\frac{1}{16}\times\frac{F_{SYS}}{Pre-scale\times(65536-\{RH3,RL3\})}$ [5]	5

[1] SM2 (SCON.5) or SM2_1(SCON_1.5) is set as logic 1.
[2] SMOD (PCON.7) or SMOD_1(T3CON.7) is set as logic 1.
[3] Timer 1 is configured as a timer in auto-reload mode (Mode 2).
[4] T1M (CKCON.4) is set as logic 1. While SMOD is 1, TH1 should not be FFH.
[5] {RH3,RL3} in the formula means $256\times RH3+RL3$. While SMOD is 1 and pre-scale is 1/1, {RH3,RL3} should not be FFFFH.

Important: Since the limitation of baud rate generator, **Suggest setting baud rate under 38400 when system timer base 16MHz HIRC value.** Following show the baud rate value table show the deviation upper 38400 baud rate.

HIRC	Target Baud Rate	RHx	RLx	RHx + RLx DEC Value	Actual Baud Rate	Error %
16MHz	2400	0xFE	0x5F	65119	2398.081535	0.079%
	4800	0xFF	0x30	65328	4807.692308	-0.160%
	9600	0xFF	0x98	65432	9615.384615	-0.160%
	19200	0xFF	0xCC	65484	19230.76923	-0.160%
	38400	0xFF	0xE6	65510	38461.53846	-0.160%
	57600	0xFF	0xEF	65519	58823.52941	**-2.124%**
	115200	0xFF	0xF7	65527	111111.1111	**3.549%**

NOTE: RHx and RLx setting value base on baud rate formula 4 (SMOD =1) or 5 .

But In most application the baud rate 115200 is a common setting value. So we provide a special function to modify HIRC to 16.6MHz. then the deviation of baud rate will be reasonable. Following table shows the error value when HIRC and timer base modified.

SADEN_1 – Slave 1 Address Mask

7	6	5	4	3	2	1	0
SADEN_1[7:0]							
R/W							

Address: BAH Reset value: 0000 0000b

Bit	Name	Description
7:0	SADEN_1[7:0]	**Slave 1 address mask** This byte is a mask byte of UART1 that contains “don’t-care” bits (defined by zeros) to form the device’s “Given” address. The don’t-care bits provide the flexibility to address one or more slaves at a time.

The following examples will help to show the versatility of this scheme.

Example 1, slave 0:

```
SADDR = 11000000b
SADEN = 11111101b
Given = 110000X0b
```

Example 2, slave 1:

```
SADDR = 11000000b
SADEN = 11111110b
Given = 1100000Xb
```

In the above example SADDR is the same and the SADEN data is used to differentiate between the two slaves. Slave 0 requires 0 in bit 0 and it ignores bit 1. Slave 1 requires 0 in bit 1 and bit 0 is ignored. A unique address for Slave 0 would be 1100 0010 since slave 1 requires 0 in bit 1. A unique address for slave 1 would be 11000001b since 1 in bit 0 will exclude slave 0. Both slaves can be selected at the same time by an address, which has bit 0 = 0 (for slave 0) and bit 1 = 0 (for slave 1). Thus, both could be addressed with 11000000b as their “Broadcast” address.

In a more complex system the following could be used to select slaves 1 and 2 while excluding slave 0:

Example 1, slave 0:

```
SADDR = 11000000b
SADEN = 11111001b
Given = 11000XX0b
```

Example 2, slave 1:

```
SADDR = 11100000b
SADEN = 11111010b
Given = 11100X0Xb
```

Figure 14.2 shows a typical interconnection of SPI devices. The bus generally connects devices together through three signal wires, MOSI to MOSI, MISO to MISO, and SPCLK to SPCLK. The Master devices select the individual Slave devices by using four pins of a parallel port to control the four $\overline{SS}$ pins. MCU1 and MCU2 play either Master or Slave mode. The $\overline{SS}$ should be configured as Master Mode Fault detection to avoid multi-master conflict.

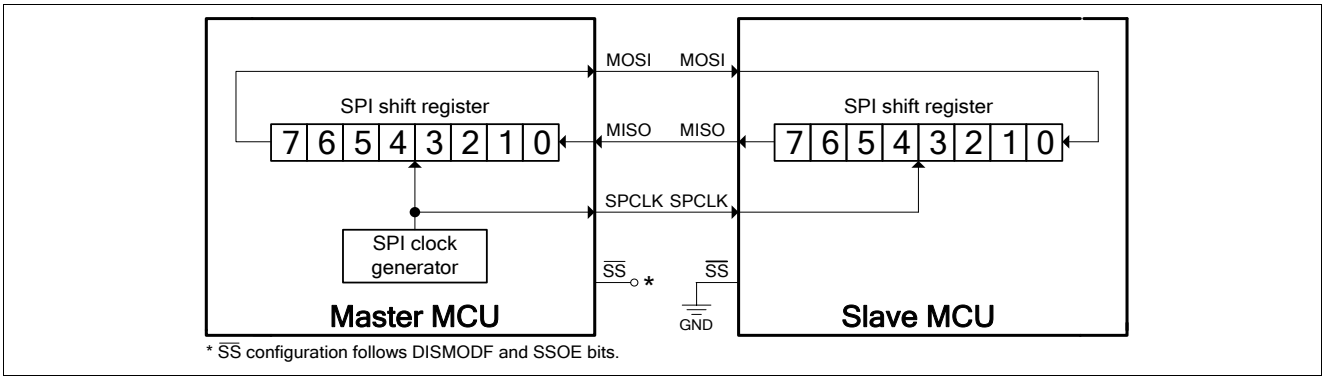

Figure 14.3. SPI Single-Master, Single-Slave Interconnection

Figure 14.3 shows the simplest SPI system interconnection, single-master and signal-slave. During a transfer, the Master shifts data out to the Slave via MOSI line. While simultaneously, the Master shifts data in from the Slave via MISO line. The two shift registers in the Master MCU and the Slave MCU can be considered as one 16-bit circular shift register. Therefore, while a transfer data pushed from Master into Slave, the data in Slave will also be pulled in Master device respectively. The transfer effectively exchanges the data, which was in the SPI shift registers of the two MCUs.

By default, SPI data is transferred MSB first. If the LSBFE (SPCR.5) is set, SPI data shifts LSB first. This bit does not affect the position of the MSB and LSB in the data register. Note that all the following description and figures are under the condition of LSBFE logic 0. MSB is transmitted and received first.

There are three SPI registers to support its operations, including SPI control register (SPCR), SPI status register (SPSR), and SPI data register (SPDR). These registers provide control, status, data storage functions, and clock rate selection. The following registers relate to SPI function.

Pin interrupt is generally used to detect an edge transient from peripheral devices like keyboard or keypad. During idle state, the system prefers to enter Power-down mode to minimize power consumption and waits for event trigger. Pin interrupt can wake up the device from Power-down mode.

PICON – Pin Interrupt Control

7	6	5	4	3	2	1	0
PIT67	PIT45	PIT3	PIT2	PIT1	PIT0	PIPS[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	

Address: E9H Reset value: 0000 0000b

Bit	Name	Description
7	PIT67	**Pin interrupt channel 6 and 7 type select** This bit selects which type that pin interrupt channel 6 and 7 is triggered. 0 = Level triggered. 1 = Edge triggered.
6	PIT45	**Pin interrupt channel 4 and 5 type select** This bit selects which type that pin interrupt channel 4 and 5 is triggered. 0 = Level triggered. 1 = Edge triggered.
5	PIT3	**Pin interrupt channel 3 type select** This bit selects which type that pin interrupt channel 3 is triggered. 0 = Level triggered. 1 = Edge triggered.
4	PIT2	**Pin interrupt channel 2 type select** This bit selects which type that pin interrupt channel 2 is triggered. 0 = Level triggered. 1 = Edge triggered.
3	PIT1	**Pin interrupt channel 1 type select** This bit selects which type that pin interrupt channel 1 is triggered. 0 = Level triggered. 1 = Edge triggered.
2	PIT0	**Pin interrupt channel 0 type select** This bit selects which type that pin interrupt channel 0 is triggered. 0 = Level triggered. 1 = Edge triggered.
1:0	PIPS[:0]	**Pin interrupt port select** This field selects which port is active as the 8-channel of pin interrupt. 00 = Port 0. 01 = Port 1. 10 = Port 2. 11 = Port 3.

Bit	Name	Description
3	PIO13	**P0.4/PWM3 pin function select** 0 = P0.4/PWM3 pin functions as P0.4. 1 = P0.4/PWM3 pin functions as PWM3 output.
2	PIO12	**P0.5/PWM2 pin function select** 0 = P0.5/PWM2 pin functions as P0.5. 1 = P0.5/PWM2 pin functions as PWM2 output.
1	PIO11	**P1.4/PWM1 pin function select** 0 = P1.4/PWM1 pin functions as P1.4. 1 = P1.4/PWM1 pin functions as PWM1 output.

17.1.2 PWM Types

The PWM generator provides two PWM types: edge-aligned or center-aligned. PWM type is selected by PWMTYP (PWMCON1.4).

PWMCON1 – PWM Control 1

7	6	5	4	3	2	1	0
PWMMOD[1:0]		GP	PWMTYP	FBINEN	PWMDIV[2:0]		
R/W		R/W	R/W	R/W	R/W		

Address: DFH Reset value: 0000 0000b

Bit	Name	Description
4	PWMTYP	**PWM type select** 0 = Edge-aligned PWM. 1 = Center-aligned PWM.

17.1.2.1 Edge-Aligned Type

In edge-aligned mode, the 16-bit counter uses single slop operation by counting up from 0000H to {PWMPH, PWMPL} and then starting from 0000H. The PWM generator signal (PGn before PWM and Fault Brake output control) is cleared on the compare match of 16-bit counter and the duty register {PWMnH, PWMnL} and set at the 16-bit counter is 0000H. The result PWM output waveform is left-edge aligned.

Examples of timed assess are shown to illustrate correct or incorrect writing process.

Example 1,

```
MOV    TA,#0AAH            ;3 clock cycles
MOV    TA,#55H             ;3 clock cycles
ORL    WDCON,#data         ;4 clock cycles
```

Example 2,

```
MOV    TA,#0AAH            ;3 clock cycles
MOV    TA,#55H             ;3 clock cycles
NOP                        ;1 clock cycle
ANL    BODCON0,#data       ;4 clock cycles
```

Example 3,

```
MOV    TA,#0AAH            ;3 clock cycles
MOV    TA,#55H             ;3 clock cycles
MOV    WDCON,#data1        ;3 clock cycles
ORL    BODCON0,#data2      ;4 clock cycles
```

Example 4,

```
MOV    TA,#0AAH            ;3 clock cycles
NOP                        ;1 clock cycle
MOV    TA,#55H             ;3 clock cycles
ANL    BODCON0,#data       ;4 clock cycles
```

In the first example, the writing to the protected bits is done before the 3-clock-cycle window closes. In example 2, however, the writing to BODCON0 does not complete during the window opening, there will be no change of the value of BODCON0. In example 3, the WDCON is successful written but the BODCON0 write is out of the 3-clock-cycle window. Therefore, the BODCON0 value will not change either. In Example 4, the second write 55H to TA completes after 3 clock cycles of the first write TA of AAH, and thus the timed access window is not opened at all, and the write to the protected byte affects nothing.

IP – Interrupt Priority (Bit-addressable)[1]

7	6	5	4	3	2	1	0
-	PADC	PBOD	PS	PT1	PX1	PT0	PX0
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: B8H Reset value: 0000 0000b

Bit	Name	Description
6	PADC	**ADC interrupt priority low bit**
5	PBOD	**Brown-out detection interrupt priority low bit**
4	PS	**Serial port 0 interrupt priority low bit**
3	PT1	**Timer 1 interrupt priority low bit**
2	PX1	**External interrupt 1 priority low bit**
1	PT0	**Timer 0 interrupt priority low bit**
0	PX0	**External interrupt 0 priority low bit**

[1] IP is used in combination with the IPH to determine the priority of each interrupt source. See Table 20-2. Interrupt Priority Level Setting for correct interrupt priority configuration.

IPH – Interrupt Priority High[2]

7	6	5	4	3	2	1	0
-	PADCH	PBODH	PSH	PT1H	PX1H	PT0H	PX0H
-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: B7H, Page0 Reset value: 0000 0000b

Bit	Name	Description
6	PADC	**ADC interrupt priority high bit**
5	PBOD	**Brown-out detection interrupt priority high bit**
4	PSH	**Serial port 0 interrupt priority high bit**
3	PT1H	**Timer 1 interrupt priority high bit**
2	PX1H	**External interrupt 1 priority high bit**
1	PT0H	**Timer 0 interrupt priority high bit**
0	PX0H	**External interrupt 0 priority high bit**

[2] IPH is used in combination with the IP respectively to determine the priority of each interrupt source. See Table 20-2. Interrupt Priority Level Setting for correct interrupt priority configuration.

EIP – Extensive Interrupt Priority[3]

7	6	5	4	3	2	1	0
PT2	PSPI	PFB	PWDT	PPWM	PCAP	PPI	PI2C
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address: EFH Reset value: 0000 0000b

Bit	Name	Description
7	PT2	**Timer 2 interrupt priority low bit**

IAP Mode	IAPCN				IAPA[15:0] {IAPAH, IAPAL}	IAPFD[7:0]
	IAPB[1:0]	FOEN	FCEN	FCTRL[3:0]		
CONFIG byte-program	11	1	0	0001	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H	Data in
CONFIG byte-read	11	0	0	0000	CONFIG0: 0000H CONFIG1: 0001H CONFIG2: 0002H CONFIG4: 0004H	Data out

[1] "X" means "don't care".
[2] Each page is 128 Bytes size. Therefore, the address should be the address pointed to the target page.

21.2 IAP User Guide

IAP facilitates the updating flash contents in a convenient way; however, user should follow some restricted laws in order that the IAP operates correctly. Without noticing warnings will possible cause undetermined results even serious damages of devices. Furthermore, this paragraph will also support useful suggestions during IAP procedures.

(1) If no more IAP operation is needed, user should clear IAPEN (CHPCON.0). It will make the system void to trigger IAP unaware. Furthermore, IAP requires the HIRC running. If the external clock source is selected, disabling IAP will stop the HIRC for saving power consumption. Note that a write to IAPEN is TA protected.

(2) When the LOCK bit (CONFIG0.1) is activated, IAP reading, writing, or erasing can still be valid.

During IAP progress, interrupts (if enabled) should be disabled temporally by clearing EA bit for implement limitation.

Do not attempt to erase or program to a page that the code is currently executing. This will cause unpredictable program behavior and may corrupt program data.

21.3 Using Flash Memory as Data Storage

In general application, there is a need of data storage, which is non-volatile so that it remains its content even after the power is off. Therefore, in general application user can read back or update the data, which rules as parameters or constants for system control. The Flash Memory array of the N76E003 supports IAP function and any byte in the Flash Memory array may be read using the MOVC instruction and thus is suitable for use as non-volatile data storage. IAP provides erase and program function that makes it easy for one or more bytes within a page to be erased and programmed in a routine. IAP performs in the application under the

24. POWER MONITORING

To prevent incorrect execution during power up and power drop, The N76E003 provide two power monitor functions, power-on detection and brown-out detection.

24.1 Power-On Reset (POR)

The power-on detection function is designed for detecting power up after power voltage reaches to a level where system can work. After power-on detected, the POF (PCON.4) will be set 1 to indicate a cold reset, a power-on reset complete. The POF flag can be cleared via software.

PCON – Power Control

7	6	5	4	3	2	1	0
SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W

Address: 87H Reset value: see Table 6-2. SFR Definitions and Reset Values

Bit	Name	Description
4	POF	**Power-on reset flag** This bit will be set as 1 after a power-on reset. It indicates a cold reset, a power-on reset complete. This bit remains its value after any other resets. This flag is recommended to be cleared via software.

Notice:

N76E003 provides power-on detection to prevent incorrect execution during power up and power drop. The power-on detection function is designed for detecting power up after power voltage reaches to a level where system can work. The N76E003 POR-detect-voltage stables at one value which falls between 1.3 V to 1.5 V. When N76E003 runs in power down mode, the core runs under a low power consumption condition. Every time N76E003 wakes up from power-down mode, power consumption condition changes to normal power consumption. It can cause the core voltage glitch to less than 1.5 V. The POR will be trigger, and MCU will reset.

Workaround:

POR is for use by VDD power-on. After power-on, the system uses LVR for power detection. Strongly suggests that disable POR function every time after power-on reset at the initial part of Customer code.

To disable POR:

At SFR address, FDH is the PORDIS register to control disable POR function through software.