SH79F3218

8051 Microcontroller with 8 channels Touch-key input and 12 Bit PWM

1. Features

- 8bits micro-controller with Pipe-line structured 8051 compatible instruction set
- Flash ROM: 32K Bytes (Rub/write 10000 times)
- RAM: internal RAM 256 Bytes, external 1280 Bytes, LED RAM 28 Bytes, Touch key RAM 16 Bytes
- EEPROM-like: 1024 Bytes (Rub/write 100000 times)
- Operation Voltage:
  - f_{OSC} = 2MHz - 12MHz, V_{DD} = 2.7V - 5.5V
- Oscillator (code option):
  - Crystal oscillator: 32.768kHz
  - Crystal oscillator: 2MHz - 12MHz
  - Ceramic oscillator: 2MHz - 12MHz
  - Internal RC: 16.6MHz (T_A = +25°C, ±1%), (T_A = -40°C +85°C, ±2%)
  - Internal RC: 128KHz (±10%)
- 24pin: 22 CMOS bi-directional I/O pins
- Built-in pull-up resistor for input pin (Typical value 30K)
- Seven big current driver I/O
  (7 common-cathode/common-anode LED)
- Four sink current driver I/O (P0.1 - P0.4)
- Five 16-bit timer/counters T0, T1, T2, T3, T4
- Two 12 Bit PWM
- Powerful interrupt sources:
  - Timer 0, 1, 2, 3, 4
  - INT40, INT45 - NT47
  - INT30 - INT32
  - INT2
  - EUART, TWI, SCM, LPD
  - PWM1, 2, TOUCH, CRC
- 8 channels Touch Key input
- EUART with Baud-rate generator
- TWI
- LED driver
  (7COM common-cathode/common-anode LED)
  - 7 X 13 dots
- LPD
  - LPD voltage: 2.8V - 4.2V
- Low Voltage Reset (LVR) function (enabled by code option)
  - LVR Voltage 1: 3.7V
  - LVR Voltage 2: 3.1V
  - LVR Voltage 3: 2.8V
- CPU Machine cycle: 1 oscillator clock
- Watch Dog Timer (WDT)
- Warm-up Timer
- CRC verify module (CRC)
- Support Low power operation modes:
  - Idle Mode
  - Power-Down Mode
- Flash Type
- Package:
  - SOP24

2. General Description

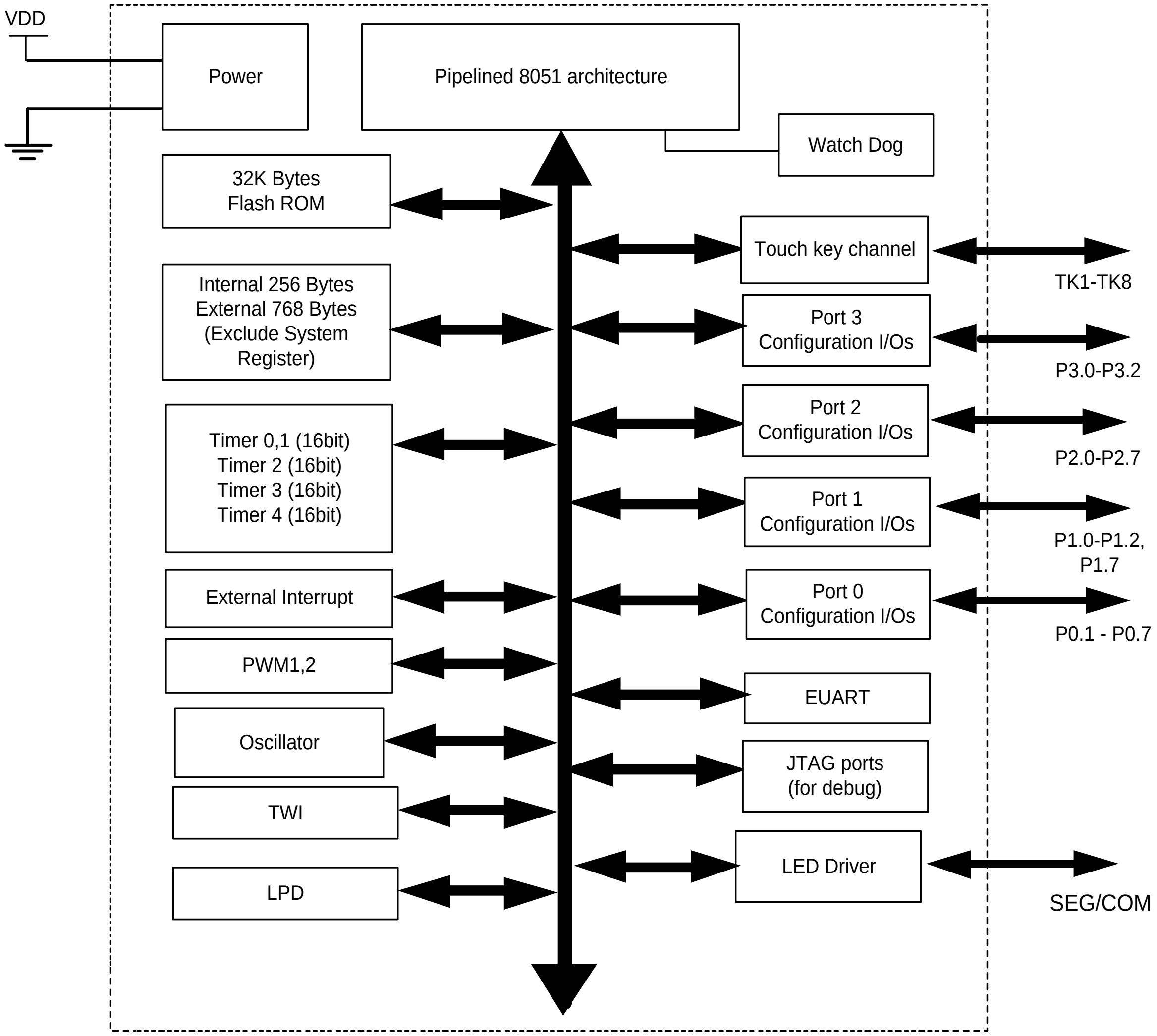
The SH79F3218 is a high performance 8051 compatible micro-controller, regard to its build-in Pipe-line instruction fetch structure, that helps the SH79F3218 can perform more fast operation speed and higher calculation performance, if compare SH79F3218 with standard 8051 at same clock speed.

The SH79F3218 retains most features of the standard 8051. These features include internal 256 bytes RAM, UART and INT2, INT3 and INT4. In addition, the SH79F3218 provides external 1280 bytes RAM, Five 16-bit timer/counters T2-T5. It also contains 32K bytes Flash memory block both for program and data.

Also the EUART, TWI, LPD, LED Driver, PWM module functions are incorporated in SH79F3218.

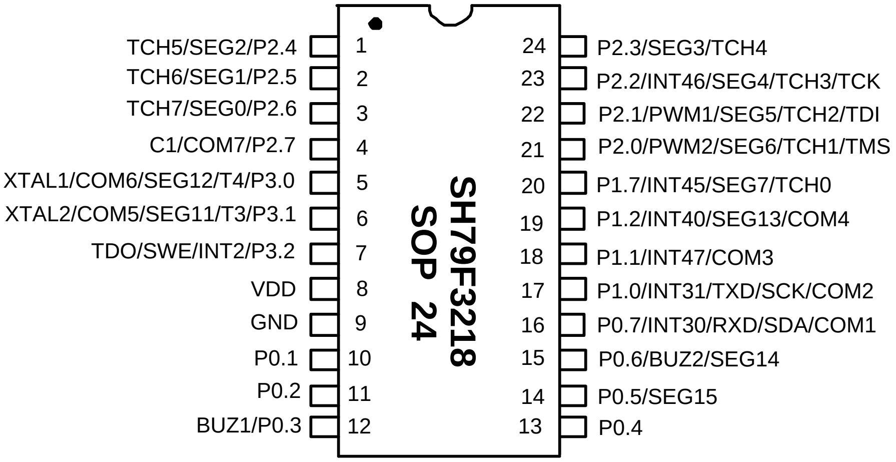
For high reliability and low power consumption, the SH79F3218 builds in Watchdog Timer, Low Voltage Reset function and SCM function. And SH79F3218 also supports two power saving modes to reduce power consumption.

3. Block Diagram



4. Pin Configuration

SOP 24



Pin Configuration Diagram

Note:

(1) The unused I/O needs to set the output (keep the low level) or input on the pull (keep the high level), avoiding the leakage and interference caused by the pin floating.
P3.3, P3.4, P3.6, P0.0 need to set. (PxPCRy, x = 0-4, y = 0-7) (Px.y, x = 0-4, y = 0-7). SH79F3218 has not ADC function.

(2) The out most pin function has the highest priority, and the inner most pin function has the lowest priority (Refer to Pin Configuration Diagram. This means when one pin is occupied by a higher priority function (if enabled) cannot be used as the lower priority functional pin, even when the lower priority function is also enabled. Until the higher priority function is closed by software, can the corresponding pin be released for the lower priority function use.

Table 4.1 Pin Function

Pin No. (SOP24)	Pin Name	Default Function
1	TCH5/SEG2/P2.4	P2.4
2	TCH6/SEG1/P2.5	P2.5
3	TCH7/SEG0/P2.6	P2.6
4	C1/COM7/P2.7	P2.7
5	XTAL1/COM5/SEG12/T4/P3.0	P3.0
6	XTAL2/COM6/SEG11/T3/P3.1	P3.1
7	SWE/INT2/P3.2	P3.2
8	VDD	---
9	GND	---
10	P0.1	P0.1
11	P0.2	P0.2
12	BUZ1/P0.3	P0.3
13	P0.4	P0.4
14	SEG15/P0.5	P0.5
15	SEG14/BUZ2/P0.6	P0.6
16	COM1/SDA/RXD/INT30/P0.7	P0.7
17	COM2/SCK/TXD/INT31/P1.0	P1.0
18	COM3/INT47/P1.1	P1.1
19	COM4/SEG13/INT40/P1.2	P1.2
20	TCH0/SEG7/INT45/P1.7	P1.7
21	TMS/TCH1/SEG6/PWM2/P2.0	P2.0
22	TDI/TCH2/SEG5/PWM1/P2.1	P2.1
23	TCK/TCH3/SEG4/INT46/P2.2	P2.2
24	TCH4/SEG3/P2.3	P2.3

Note:

The unused I/O needs to set the output (keep the low level) or input on the pull (keep the high level), avoiding the leakage and interference caused by the pin floating.

P3.3, P3.4, P3.6, P0.0 need to set. (PxPCRy, x = 0-4, y = 0-7) (Px.y, x = 0-4, y = 0-7). SH79F3218 has not ADC function.

5. Pin Description

Pin No.	Type	Description
I/O PORT		
P0.1 - P0.7	I/O	7 bit General purpose CMOS I/O
P1.0 - P1.2, P1.7	I/O	4 bit General purpose CMOS I/O
P2.0 - P2.7	I/O	8 bit General purpose CMOS I/O
P3.0 - P3.2	I/O	3 bit General purpose CMOS I/O
Timer		
T3	I	Timer3 extemal input
T4	I/O	Timer4 external input/output
PWM		
PWM1	O	Output pin for 12-bit PWM1 timer
PWM2	O	Output pin for 12-bit PWM2 timer
EUART		
RXD	I	EUART data input
TXD	O	EUART data output
LED		
LED_C1 - LED_C7	O	Common signal output for LED display
LED_S0 - S7 LED_S11 - S15	O	Segment signal output for LED display
IIC		
SDA	I/O	TWI data input/output
SCK	I/O	TWI clock
TK		
TCH0 - 7	I	Touch Key pin
BUZZER		
BUZ1, 2	O	Buzzer output pin
Interrupt & Reset & Clock & Power		
INT2	I	External interrupt 2 input source
INT30	I	External interrupt 30 input source
INT31	I	External interrupt 31 input source
INT40, INT45 - 47	I	External interrupt 40, 45-47 input source
XTAL1	I	Oscillator input
XTAL2	O	Oscillator output
GND	P	Ground
VDD	P	Power supply (2.7 - 5.5V)

(continued)

Pin No.	Type	Description
Programmer		
SWE (P3.2)	I/O	One line Debug interface
TDO (P3.2)	O	Debug interface: Test data out
TCK (P2.2)	I	Debug interface: Test mode select
TDI (P2.1)	I	Debug interface: Test data in
TMS (P2.0)	I	Debug interface: Test clock in

Note:

When P3.2，P2.2 - P2.0 used as debug interface, functions of P3.2，P2.2 - P2.0 are blocked.
When P3.2 used as one line debug interface, functions of P3.2 are blocked.

6. SFR Mapping

The SH79F3218 provides 256 bytes of internal RAM to contain general-purpose data memory and Special Function Register (SFR). The SFR of the SH79F3218 fall into the following categories:

CPU Core Registers:	ACC, B, PSW, SP, DPL, DPH
Enhanced CPU Core Registers:	AUXC, DPL1, DPH1, INSCON, XPAGE
Power and Clock Control Registers:	PCON, SUSLO
Flash Registers:	IB_OFFSET, IB_DATA, IB_CON1, IB_CON2, IB_CON3, IB_CON4, IB_CON5, FLASHCON
Data Memory Register:	XPAGE
Watchdog Timer Registers:	RSTSTAT
System Clock Control Register:	CLKCON
Interrupt System Registers:	IEN0, IEN1, IENC, IENC1, IENC2, IPH0, IPL0, IPH1, IPL1, EXF0, EXF1, EXF3, EXCON0, EXCON1, EXCON2
I/O Port Registers:	P0, P1, P2, P3, P4, P0CR, P1CR, P2CR, P3CR, P4CR, P0PCR, P1PCR, P2PCR, P3PCR, P4PCR, P1OS
Timer Registers:	TCON, TMOD, TL0, TH0, TL1, TH1, TCON1, T2CON, T2MOD, TH2, TL2, RCAP2L, RCAP2H, T3CON, TH3, TL3, T4CON, TH4, TL4, SWTHL
EUART Registers:	SCON, SBUF, SADEN, SADDR, PCON, RxCON
LED Registers:	DISPCON, SEG01, SEG02, DISPCLK, LEDCOM, DISCOM, LIGHTCOM, SHARECON
BUZZER Registers:	BUZCON, BUZD, BUZP
TWI Registers:	TWIDAT, TWIADR, TWISTA, TWICON
PWM Registers:	PWMEN, PWMEN1, PWMLO, PWM1C, PWM2C, PWM1PL, PWM1PH, PWM1DL, PWM1DH, PWM2C, PWM2PL, PWM2PH, PWM2DL, PWM2DH
LPD Registers:	LPDCON
TK Registers:	TKCON1, TKF0, TKU1, TKDIV01, TKDIV02, TKDIV03, TKDIV04, TKVREF, TKST, TKRANDOM, TKCOUNT, TKW, P2SS, TKWAIT
CRC Registers:	CRCCON, CRCDL, CRCDH

Table 6.1 CPU Core SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ACC	E0H	Accumulator	00000000	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
B	F0H	B Register	00000000	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
AUXC	F1H	C Register	00000000	C.7	C.6	C.5	C.4	C.3	C.2	C.1	C.0
PSW	D0H	Program Status Word	00000000	CY	AC	F0	RS1	RS0	OV	F1	P
SP	81H	Stack Pointer	00000111	SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0
DPL	82H	Data Pointer Low byte	00000000	DPL0.7	DPL0.6	DPL0.5	DPL0.4	DPL0.3	DPL0.2	DPL0.1	DPL0.0
DPH	83H	Data Pointer High byte	00000000	DPH0.7	DPH0.6	DPH0.5	DPH0.4	DPH0.3	DPH0.2	DPH0.1	DPH0.0
DPL1	84H	Data Pointer 1 Low byte	00000000	DPL1.7	DPL1.6	DPL1.5	DPL1.4	DPL1.3	DPL1.2	DPL1.1	DPL1.0
DPH1	85H	Data Pointer 1 High byte	00000000	DPH1.7	DPH1.6	DPH1.5	DPH1.4	DPH1.3	DPH1.2	DPH1.1	DPH1.0
INSCON	86H	Data pointer select	-0--00-0	-	BKS0	-	-	DIV	MUL	-	DPS

Table 6.2 Power and Clock control SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	87H	Power Control	00--0000	SMOD	SSTAT	-	-	GF1	GF0	PD	IDL
SUSLO	8EH	Suspend Mode Control	00000000	SUSLO.7	SUSLO.6	SUSLO.5	SUSLO.4	SUSLO.3	SUSLO.2	SUSLO.1	SUSLO.0

Table 6.3 Flash control SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_OFF SET	FBH Bank0	Low byte offset of flash memory for programming	00000000	IB_OFF SET.7	IB_OFF SET.6	IB_OFF SET.5	IB_OFF SET.4	IB_OFF SET.3	IB_OFF SET.2	IB_OFF SET.1	IB_OFF SET.0
IB_DATA	FCH Bank0	Data Register for programming flash memory	00000000	IB_DATA.7	IB_DATA.6	IB_DATA.5	IB_DATA.4	IB_DATA.3	IB_DATA.2	IB_DATA.1	IB_DATA.0
IB_CON1	F2H Bank0	Flash Memory Control Register 1	00000000	IB_CON1.7	IB_CON1.6	IB_CON1.5	IB_CON1.4	IB_CON1.3	IB_CON1.2	IB_CON1.1	IB_CON1.0
IB_CON2	F3H Bank0	Flash Memory Control Register 2	----0000	-	-	-	-	IB_CON2.3	IB_CON2.2	IB_CON2.1	IB_CON2.0
IB_CON3	F4H Bank0	Flash Memory Control Register 3	----0000	-	-	-	-	IB_CON3.3	IB_CON3.2	IB_CON3.1	IB_CON3.0
IB_CON4	F5H Bank0	Flash Memory Control Register 4	----0000	-	-	-	-	IB_CON4.3	IB_CON4.2	IB_CON4.1	IB_CON4.0
IB_CON5	F6H Bank0	Flash Memory Control Register 5	----0000	-	-	-	-	IB_CON5.3	IB_CON5.2	IB_CON5.1	IB_CON5.0
XPAGE	F7H Bank0	Memory Page	00000000	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
FLASHCON	A7H Bank0	Flash access control	-------0	-	-	-	-	-	-	-	FAC

Table 6.4 WDT SFR

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RSTSTAT	B1H Bank0	Watchdog Timer Control	0000-000*	WDOF	SWRF	PORF	LVRF	-	WDT.2	WDT.1	WDT.0

***Note:** *RSTSTAT initial value is determined by different RESET, refer to “Watchdog Timer (WDT)” section for details.*

Table 6.5 CLKCON SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	B2H Bank0	System Clock Control Register	111000--	32k_SPDUP	CLKS1	CLKS0	SCMIF	HFON	FS	-	-

Table 6.6 Interrupt SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	A8H Bank0	Interrupt Enable Control 0	0-000000	EA	-	ET2	ES	ET1	EPWM2	ET0	TKIE
IEN1	A9H Bank0	Interrupt Enable Control 1	00000000	ESCM	EX3	EPWM1	ET3	ETWI	EX4	EX2	ET4
IENC	BAH Bank0	Interrupt 4channel enable control	000----0	EXS47	EXS46	EXS45	-	-	-	-	EXS40
IENC1	BBH Bank0	Interrupt channel enable control 1	--000-00	-	-	ECRC	EPWM1	EPWM2	-	ESCM1	ELPD
IENC2	BCH Bank0	Interrupt channel enable control 2	-----000	-	-	-	-	-	EXS32	EXS31	EXS30
IPH0	B4H Bank0	Interrupt Priority Control High 0	--000000	-	-	PT2H	PESH	PX3H	PW2H	PT0H	PTKH
IPL0	B8H Bank0	Interrupt Priority Control Low 0	--000000	-	-	PT2L	PESL	PX3L	PW2L	PTKL	PT0L
IPH1	B5H Bank0	Interrupt Priority Control High 1	00000000	PSCH	PT1H	PW1H	PT3H	PTWH	PX4H	PX2H	PT4H
IPL1	B9H Bank0	Interrupt Priority Control Low 1	00000000	PSCL	PT1L	PW1L	PT3L	PTWL	PX4L	PX2L	PT4L
EXF0	E8H Bank0	External interrupt Control 0	00000000	IT4.1	IT4.0	IT3.1	IT3.0	IT2.1	IT2.0	IE3	IE2
EXF1	D8H Bank0	External interrupt Control 1	000----0	IF47	IF46	IF45	-	-	-	-	IF40
EXF3	D1H Bank0	External interrupt Control 3	------00	-	-	-	-	-	-	IF31	IF30
EXCON	A4H Bank0	External interrupt Sample Control	00000000	EXCON	I1PS1	I1PS0	I1SN1	I1SN0	I0PS1	I0PS0	I0SN1

Table 6.7 Port SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0	80H Bank0	8-bit Port 0	00000000	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P1	90H Bank0	8-bit Port 1	00000000	P1.7	*	*	*	*	P1.2	P1.1	P1.0
P2	A0H Bank0	8-bit Port 2	00000000	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P3	B0H Bank0	8-bit Port 3	00000000	*	P3.6	*	P3.4	P3.3	P3.2	P3.1	P3.0
P4	90H Bank1	8-bit Port 4	00000000	*	*	*	*	*	*	*	*
P0CR	E1H Bank0	Port0 input/output direction control	00000000	P0CR.7	P0CR.6	P0CR.5	P0CR.4	P1CR.3	P1CR.2	P1CR.1	P1CR.0
P1CR	E2H Bank0	Port1 input/output direction control	00000000	P1CR.7	*	*	*	*	P1CR.2	P1CR.1	P1CR.0
P2CR	E3H Bank0	Port2 input/output direction control	00000000	P2CR.7	P2CR.6	P2CR.5	P2CR.4	P2CR.3	P2CR.2	P2CR.1	P2CR.0
P3CR	E4H Bank0	Port3 input/output direction control	00000000	*	P3CR.6	*	P3CR.4	P3CR.3	P3CR.2	P3CR.1	P3CR.0
P4CR	91H Bank1	Port4 input/output direction control	00000000	*	*	*	*	*	*	*	*
P0PCR	E9H Bank0	Internal pull-high enable for Port0	00000000	P0PCR.7	P0PCR.6	P0PCR.5	P0PCR.4	P0PCR.3	P0PCR.2	P0PCR.1	P0PCR.0
P1PCR	EAH Bank0	Internal pull-high enable for Port1	00000000	P1PCR.7	*	*	*	*	P1PCR.2	P1PCR.1	P1PCR.0
P2PCR	EBH Bank0	Internal pull-high enable for Port2	00000000	P2PCR.7	P2PCR.6	P2PCR.5	P2PCR.4	P2PCR.3	P2PCR.2	P2PCR.1	P2PCR.0
P3PCR	ECH Bank0	Internal pull-high enable for Port3	00000000	*	P3PCR.6	*	P3PCR.4	P3PCR.3	P3PCR.2	P3PCR.1	P3PCR.0
P4PCR	92H Bank1	Internal pull-high enable for Port4	00000000	*	*	*	*	*	*	*	*
P1OS	EFH Bank0	Output mode control	----0000	-	-	-	-	P1OSCR.1	P1OSCR.0	P1OS.1	P1OS.0
P2SS	9DH Bank0	Touch key mode control	00000000	P2SS.7	P2SS.6	P2SS.5	P2SS.4	P2SS.3	P2SS.2	P2SS.1	P2SS.0

***Note:** *Please refer to the 7.7 I/O port section (P34).*

Table 6.8 Timer SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	88H Bank0	Timer/Counter 0/1 Control	0000----	TF1	TR1	TF0	TR0	-	-	-	-
TMOD	89H Bank0	Timer/Counter 0/1 Mode	--00--00	-	-	M11	M10	-	-	M01	M00
TL0	8AH Bank0	Timer/Counter 0 Low Byte	00000000	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
TH0	8BH Bank0	Timer/Counter 0 High Byte	00000000	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
TL1	8CH Bank0	Timer/Counter 1 Low Byte	00000000	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
TH1	8DH Bank0	Timer/Counter 1 High Byte	00000000	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
TCON1	8FH Bank0	Timer/Counter 0/1 Mode Control	-00-00--	-	TCLK_S1	TCLK_S0	-	TCLKP1	TCLKP0	-	-
T2CON	C8H Bank0	Timer/Counter 2 Control	0----0--	TF2	-	-	-	-	TR2	-	-
T2MOD	C9H Bank0	Timer/Counter 2 Mode	0--------	TCLKP2	-	-	-	-	-	-	-
RCAP2L	CAH Bank0	Timer/Counter 2 Reload /Capture Low Byte	00000000	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
RCAP2H	CBH Bank0	Timer/Counter 2 Reload /Capture High Byte	00000000	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
TL2	CCH Bank0	Timer/Counter 2 Low Byte	00000000	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
TH2	CDH Bank0	Timer/Counter 2 High Byte	00000000	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
T3CON	C0H Bank1	Timer/Counter 3 Control	0-00-000	TF3	-	T3PS.1	T3PS.0	-	TR3	T3CLKS.1	T3CLKS.0
TL3	C2H Bank1	Timer/Counter 3 Low Byte	00000000	TL3.7	TL3.6	TL3.5	TL3.4	TL3.3	TL3.2	TL3.1	TL3.0
TH3	C3H Bank1	Timer/Counter 3 High Byte	00000000	TH3.7	TH3.6	TH3.5	TH3.4	TH3.3	TH3.2	TH3.1	TH3.0
T4CON	C8H Bank1	Timer/Counter 4 Control	00000000	TF4	TC4	T4PS1	T4PS0	T4M1	T4M0	TR4	T4CLKS
TL4	CCH Bank1	Timer/Counter 4 Low Byte	00000000	TL4.7	TL4.6	TL4.5	TL4.4	TL4.3	TL4.2	TL4.1	TL4.0
TH4	CDH Bank1	Timer/Counter 4 High Byte	00000000	TH4.7	TH4.6	TH4.5	TH4.4	TH4.3	TH4.2	TH4.1	TH4.0

Table 6.9 EUART SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON	98H Bank0	EUART Serial Control	00000000	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI
SBUF	99H Bank0	EUART Serial Data Buffer	00000000	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
SADEN	9BH Bank0	EUART Slave Address Mask	00000000	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
SADDR	9AH Bank0	EUART Slave Addres	00000000	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
SBRTH	C7H Bank0	EUART Baudrate Generator	00000000	SBRTEN	SBRT.14	SBRT.13	SBRT.12	SBRT.11	SBRT.10	SBRT.9	SBRT.8
SBRTL	BFH Bank0	EUART Baudrate Generator	00000000	SBRT.7	SBRT.6	SBRT.5	SBRT.4	SBRT.3	SBRT.2	SBRT.1	SBRT.0
SFINE	BEH Bank0	EUART Baudrate Generator	----0000	-	-	-	-	SFINE.3	SFINE.2	SFINE.1	SFINE.0
PCON	87H Bank0	Power & serial Control	00--0000	SMOD	SSTAT	**-**	-	GF1	GF0	PD	IDL

Table 6.10 BUZZER SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZCON	BDH Bank0	Buzzer Output Control	00--00--	BUZEN	BUSLE	-	-	BCA1	BCA0	-	-
BUZP	AFH Bank0	Buzzer Period Control	00000000	BZP.7	BZP.6	BZP.5	BZP.4	BZP.3	BZP.2	BZP.1	BZP.0

Table 6.11 CRC SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CRCCON	ACH Bank0	CRC verify control	00-00000	CRC_GO	CRCIF	-	CRCADDR4	CRCADDR3	CRCADDR2	CRCADDR1	DAT.0
CRCDL	ADH Bank0	CRC verify result low byte	00000000	CRCD.7	CRCD.6	CRCD.5	CRCD.4	CRCD.3	CRCD.2	CRCD.1	CRCD.0
CRCDH	AEH Bank0	CRC verify result high byte	00000000	CRCD.15	CRCD.14	CRCD.13	CRCD.12	CRCD.11	CRCD.10	CRCD.9	CRCD.8

Table 6.12 TK SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCON1	A1H Bank0	Touch Key Control	0-000000	TKCON	-	TKGO/$\overline{\text{DONE}}$	SHARE	MODE	OVDD	FSW1	FSW0
TKF0	C1H Bank0	Touch Key interrupt flag Register	-00000--	-	IFERR	IFGO	IFAVE	IFCOUNT	IFTKOV	-	-
TKU1	C2H Bank0	Touch Key channel selection Register	00000000	TK8	TK7	TK6	TK5	TK4	TK3	TK2	TK1
TKDIV01	C3H Bank0	Touch Key amplification coefficient Register	00000000	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
TKDIV02	C4H Bank0	Touch Key amplification coefficient Register	00000000	DIV15	DIV14	DIV13	DIV12	DIV11	DIV10	DIV9	DIV8
TKDIV03	C5H Bank0	Touch Key amplification coefficient Register	00000000	DIV23	DIV22	DIV21	DIV20	DIV19	DIV18	DIV17	DIV16
TKDIV04	C6H Bank0	Touch Key amplification coefficient Register	----0000	-	-	-	-	DIV27	DIV26	DIV25	DIV24
TKVREF	CEH Bank0	Reference voltage source selection Register	00000000	VREF1	VREF0	CMPD1	CMPD0	VTK1	VTK0	TUNE1	TUNE0
TKST	CFH Bank0	Touch Key frequency selection Register	-0000000	-	ST.6	ST.5	ST.4	ST.3	ST.2	ST.1	ST.0
TKRANDOM	B6H Bank0	Touch Key frequency selection Register	0000--00	TKRADON	TKOFFSET	TKVDD	TKOUT	-	-	RANDOM1	RANDOM0
TKW	B7H Bank0	Touch Key channel error display Register	-----000	-	-	-	-	-	TW.2	TW.1	TW.0
TKCOUNT	E6H Bank0	Touch Key clock width selection Register	00000000	COUNT0.7	COUNT0.6	COUNT0.5	COUNT0.4	COUNT0.3	COUNT0.2	COUNT0.1	COUNT0.0
TKWAIT	A2H Bank0	Touch Key Waiting clock width selection Register	00000000	TKWAIT.7	TKWAIT.6	TKWAIT.5	TKWAIT.4	TKWAIT.3	TKWAIT.2	TKWAIT.1	TKWAIT.0

Table 6.13 LPD SFR

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPDCON	B3H Bank0	LPD Control	00---000	LPDEN	LPDF	-	-	-	LPDOP	LPDS1	LPDS0

Table 6.14 LED SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DISPCON	D9H Bank0	LED Control	-0-000--	-	LEDON	-	MODSW	LEDMODE0	LEDMODE1	-	-
SEG01	DAH Bank0	SEG function selection Register	00000000	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
SEG02	DBH Bank0	SEG function selection Register	-0000---	-	SEG14	SEG13	SEG12	SEG11	-	-	-
DISPCLK	DCH Bank0	LED clock frequency selection Register	00000000	DCK0.7	DCK0.6	DCK0.5	DCK0.4	DCK0.3	DCK0.2	DCK0.1	DCK0.0
LEDCOM	DDH Bank0	COM function selection Register	-0000000	-	COM7	COM6	COM5	COM4	COM3	COM2	COM1
DISCOM	DEH Bank0	LED COM sweep length Register	00000000	DCOM.7	DCOM.6	DCOM.5	DCOM.4	DCOM.3	DCOM.2	DCOM.1	DCOM.0
LIGHTCOM	DFH Bank0	LED COM brightness selection Register	-----000	-	-	-	-	-	CC3	CC2	CC1
SHARECON	D7H Bank0	SHARE Control Register	-----000	-	-	-	-	-	SHARE.2	SHARE.1	SHARE.0

Table 6.15 TWI SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TWICON	F8H Bank0	TWI setting Register	00000000	TOUT	ENTWI	STA	STO	TWINT	AA	TFREE	EFREE
TWISTA	F9H Bank0	TWI state Register	11111000	TWISTA.7	TWISTA.6	TWISTA.5	TWISTA.4	TWISTA.3	CR.1	CR.0	ETOT
TWIADR	FAH Bank0	TWI data address Register	00000000	TWA.6	TWA.5	TWA.4	TWA.3	TWA.2	TWA.1	TWA.0	GC
TWIDAT	FDH Bank0	TWI data input/output Register	00000000	TWIDAT.7	TWIDAT.6	TWIDAT.5	TWIDAT.4	TWIDAT.3	TWIDAT.2	TWIDAT.1	TWIDAT.0

Table 6.16 PWM SFRs

符号	地址	名称	POR/WDT/LVR /PIN复位值	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
PWM1CON	E8H Bank1	PWM1 timer enable	0000-000	PWM1EN	PWM1S	PWM1CK1	PWM1CK0	-	PWM1IE	PWM1IF	PWM1SS
PWM2CON	E9H Bank1	PWM2 timer enable	0000-000	PWM2EN	PWM2S	PWM2CK1	PWM2CK0	-	PWM2IE	PWM2IF	PWM2SS
PWM1PH	EAH Bank1	12-bit PWM1 Period Control low byte	----0000	-	-	-	-	PWM1P.11	PWM1P.10	PWM1P.9	PWM1P.8
PWM1PL	EBH Bank1	12-bit PWM1 Period Control high byte	00000000	PWM1P.7	PWM1P.6	PWM1P.5	PWM1P.4	PWM1P.3	PWM1P.2	PWM1P.1	PWM1P.0
PWM2PH	ECH Bank1	12-bit PWM2 Period Control low byte	----0000	-	-	-	-	PWM2P.11	PWM2P.10	PWM2P.9	PWM2P.8
PWM2PL	EDH Bank1	12-bit PWM2 Period Control high byte	00000000	PWM2P.7	PWM2P.6	PWM2P.5	PWM2P.4	PWM2P.3	PWM2P.2	PWM2P.1	PWM2P.0
PWM1DH	E4H Bank1	12-bit PWM1 Duty Control low byte	----0000	-	-	-	-	PWM1D.11	PWM1D.10	PWM1D.9	PWM1D.8
PWM1DL	E5H Bank1	12-bit PWM1 Duty Control high byte	00000000	PWM1D.7	PWM1D.6	PWM1D.5	PWM1D.4	PWM1D.3	PWM1D.2	PWM1D.1	PWM1D.0
PWM2DH	E6H Bank1	12-bit PWM2 Duty Control low byte	----0000	-	-	-	-	PWM2D.11	PWM2D.10	PWM2D.9	PWM2D.8
PWM2DL	E7H Bank1	12-bit PWM2 Duty Control high byte	00000000	PWM2D.7	PWM2D.6	PWM2D.5	PWM2D.4	PWM2D.3	PWM2D.2	PWM2D.1	PWM2D.0

***Note:** - : Unimplemented*

SFR Map

Bank0

	Bit addressable	**Non Bit addressable**							
	0/8	*1/9*	*2/A*	*3/B*	*4/C*	*5/D*	*6/E*	*7/F*	
F8H	TWICON	TWISTA	TWIADR	IB_OFFSET	IB_DATA	TWIDAT			FFH
F0H	B	AUXC	IB_CON1	IB_CON2	IB_CON3	IB_CON4	IB_CON5	XPAGE	F7H
E8H	EXF0	P0PCR	P1PCR	P2PCR	P3PCR			P1OS	EFH
E0H	ACC	P0CR	P1CR	P2CR	P3CR		TKCOUNT		E7H
D8H	EXF1	DISPCON	SEG01	SEG02	DISPCLK	LEDCOM	DISCOM	LIGHTCOM	DFH
D0H	PSW	EXF3						SHARECON	D7H
C8H	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	TKVREF	TKST	CFH
C0H	HLVCON	TKF0	TKU1	TKDIV01	TKDIV02	TKDIV03	TKDIV04	SBRTH	C7H
B8H	IPL0	IPL1	IENC	IENC1	IENC2	BUZCON	SFINE	SBRTL	BFH
B0H	P3	RSTSTAT	CLKCON	LPDCON	IPH0	IPH1	TKRANDOM	TKW	B7H
A8H	IEN0	IEN1	ISPLO	ISPCON	CRCCON	CRCDL	CRCDH	BUZP	AFH
A0H	P2	TKCON1	TKWAIT		EXCON			FLASHCON	A7H
98H	SCON	SBUF	SADDR	SADEN		P2SS			9FH
90H	P1								97H
88H	TCON	TMOD	TL0	TH0	TL1	TH1	SUSLO	TCON1	8FH
80H	P0	SP	DPL	DPH	DPL1	DPH1	INSCON	PCON	87H
	0/8	*1/9*	*2/A*	*3/B*	*4/C*	*5/D*	*6/E*	7/F	

Bank1

	Bit addressable	**Non Bit addressable**							
	0/8	*1/9*	*2/A*	*3/B*	*4/C*	*5/D*	*6/E*	*7/F*	
F8H									FFH
F0H	B	AUXC						XPAGE	F7H
E8H	PWM1CON	PWM2CON	PWM1PH	PWM1PL	PWM2PH	PWM2PL			EFH
E0H	ACC				PWM1DH	PWM1DL	PWM2DH	PWM2DL	E7H
D8H									DFH
D0H	PSW								D7H
C8H	T4CON				TL4	TH4			CFH
C0H	T3CON		TL3	TH3					C7H
B8H	IPL0	IPL1							BFH
B0H					IPH0	IPH1			B7H
A8H	IEN0	IEN1							AFH
A0H									A7H
98H									9FH
90H	P4	P4CR	P4PCR						97H
88H							SUSLO		8FH
80H		SP	DPL	DPH	DPL1	DPH1	INSCON	PCON	87H
	0/8	*1/9*	*2/A*	*3/B*	*4/C*	*5/D*	*6/E*	7/F	

Note: *The unused addresses of SFR are not available.*

7. Normal Function

7.1 CPU

7.1.1 CPU Core SFR

Feature

- CPU core registers: ACC, B, PSW, SP, DPL, DPH

Accumulator

ACC is the Accumulator register. Instruction system adopts A as mnemonic symbol of accumulator.

B Register

The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

Stack Pointer (SP)

The Stack Pointer Register is 8 bits special register, It is incremented before data is stored during PUSH, CALL executions and it is decremented after data is out of stack during POP, RET, RETI executions. The stack may reside anywhere in on-chip internal RAM (00H-FFH). On reset, the Stack Pointer is initialized to 07H causing the stack to begin at location 08H.

Program Status Word Register (PSW)

The PSW register contains program status information.

Data Pointer Register (DPTR)

DPTR consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address, but it may be manipulated as a 16-bit register or as two independent 8-bit registers.

Table 7.1 PSW Register

D0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset Value (POR/WDT/LVR/PIN))	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**CY**	**Carry flag bit** 0: no carry or borrow in an arithmetic or logic operation 1: a carry or borrow in an arithmetic or logic operation
6	**AC**	**Auxiliary Carry flag bit** 0: no auxiliary carry or borrow in an arithmetic or logic operation 1: an auxiliary carry or borrow in an arithmetic or logic operation
5	**F0**	**F0 flag bit** Available to the user for general purposes
4-3	**RS[1:0]**	**R0-R7 Register bank select bits** 00: Bank0 (Address to 00H-07H) 01: Bank1 (Address to 08H-0FH) 10: Bank2 (Address to 10H-17H) 11: Bank3 (Address to 18H-1FH)
2	**OV**	**Overflow flag bit** 0: no overflow happen 1: an overflow happen
1	**F1**	**F1 flag bit** Available to the user for general purposes
0	**P**	**Parity flag bit** 0: In the Accumulator,the bits whose value is 1 is even number 1: In the Accumulator,the bits whose value is 1 is odd number

7.1.2 Enhanced CPU core SFRs

- Extended 'MUL' and 'DIV' instructions: 16bit*8bit, 16bit/8bit
- Dual Data Pointer
- Enhanced CPU core registers: AUXC, DPL1, DPH1, INSCON

The SH79F3218 has modified 'MUL' and 'DIV' instructions. These instructions support 16 bit operand. A new register - the register AUXC is applied to hold the upper part of the operand/result.

The AUXC register is used during 16 bit operand multiply and divide operations. For other instructions it can be treated as another scratch pad register.

After reset, the CPU is in standard mode, which means that the 'MUL' and 'DIV' instructions are operating like the standard 8051 instructions. To enable the 16 bit mode operation, the corresponding enable bit in the INSCON register must be set.

	Operation		Result		
			A	B	AUXC
MUL	INSCON.2 = 0; 8 bit mode	(A)*(B)	Low Byte	High Byte	---
	INSCON.2 = 1; 16 bit mode	(AUXC A)*(B)	Low Byte	Middle Byte	High Byte
DIV	INSCON.3 = 0; 8 bit mode	(A)/(B)	Quotient Low Byte	Remainder	---
	INSCON.3 = 1; 16 bit mode	(AUXC A)/(B)	Quotient Low Byte	Remainder	Quotient High Byte

Dual Data Pointer

Using two data pointers can accelerate data memory moves. The standard data pointer is called DPTR and the new data pointer is called DPTR1.

DPTR1 is similar to DPTR, which consists of a high byte (DPH1) and a low byte (DPL1). Its intended function is to hold a 16-bit address, but it may be manipulated as a 16-bit register or as two independent 8-bit registers.

The DPS bit in INSTCON register is used to choose the active pointer by setting 1 or 0. And all DPTR-related instructions will use the currently selected data pointer.

7.1.3 Register

Table 7.2 Data Pointer Select Register

86H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INSCON	-	BKS0	-	-	DIV	MUL	-	DPS
R/W	-	R/W	-	-	R/W	R/W	-	R/W
Reset Value (POR/WDT/LVR/PIN)	-	0	-	-	0	0	-	0

Bit Number	Bit Mnemonic	Description
6	**BKS0**	**SFR Bank Selection Bit** 0: SFR Bank0 selected 1: SFR Bank1 selected
3	**DIV**	**16 bit/8 bit Divide Selection Bit** 0: 8 bit Divide 1: 16 bit Divide
2	**MUL**	**16 bit/8 bit Multiply Selection Bit** 0: 8 bit Multiply 1: 16 bit Multiply
0	**DPS**	**Data Pointer Selection Bit** 0: Data pointer 1: Data pointer1

7.2 RAM

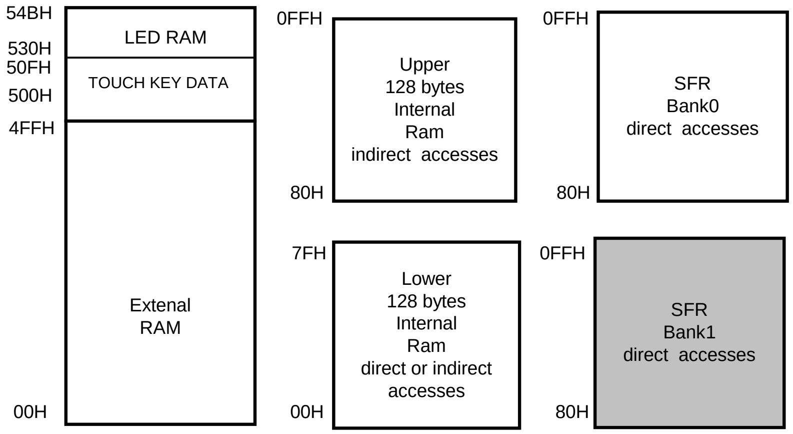
7.2.1 Features

SH79F3218 provides both internal RAM and external RAM for random data storage. The internal data memory is mapped into four separated segments:

- The Lower 128 bytes of RAM (addresses 00H to 7FH) are directly and indirectly addressable.
- The Upper 128 bytes of RAM (addresses 80H to FFH) are indirectly addressable only.
- The Special Function Registers (SFR, addresses 80H to FFH) are directly addressable only.
- The 1280 bytes of external RAM(addresses 000H to 4FFH) are indirectly accessed by MOVX instructions.

The Upper 128 bytes occupy the same address space as SFR, but they are physically separate from SFR space. When an instruction accesses an internal location above address 7FH, the CPU can distinguish whether to access the upper 128 bytes data RAM or to access SFR by different addressing mode of the instruction.

SH79F3218 provides an extra 1280 bytes of RAM to support high-level language in external data space. Also, SH79F3218 provides 28 bytes LED RAMand 16 bytes touch key RAM.



The Internal and External RAM Configuration

The SH79F3218 provides traditional method for accessing of external RAM. Use *MOVXA, @Ri* or *MOVX @Ri, A;* to access external low 256 bytes RAM; *MOVX A, @DPTR* or *MOVX @DPTR, A* also to access external 1280 bytes RAM.

In SH79F3218 the user can also use XPAGE register to access external RAM only with *MOVX A, @Ri* or *MOVX @Ri, A* instructions. The user can use XPAGE to represent the high byte address of RAM above 256 Bytes.

In Flash SSP mode, the XPAGE can also be used as sector selector (Refer to SSP Function).

7.2.2 Register

Table 7.3 Data Memory Page Register

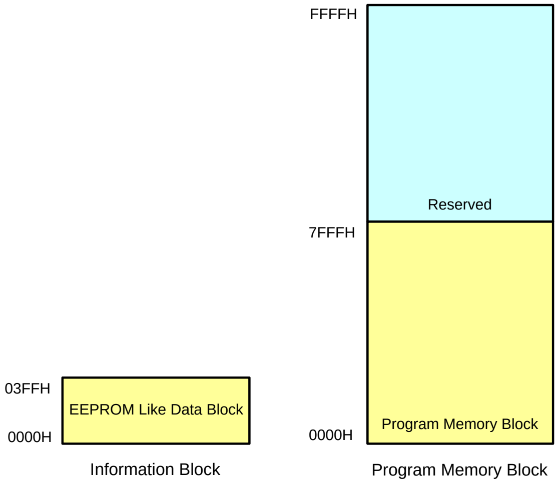
F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	-	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
6-0	**XPAGE[6:0]**	RAM Page Selector

7.3 Flash Program Memory

7.3.1 Features

- The program memory consists 32 X 1KB sectors, total 32KB
- 4 X 256 Bytes EEPROM-Like Built-in, total 1KB
- Programming and erase can be done over the full operation voltage range
- Supports 4 kinds of code protection
- Write, read and erase operation are all supported by In-Circuit Programming (ICP)
- Support overall/sector erase and programming
- Minimum program/erase cycles:
  Main program memory: 10,000
  EEPROM like memory: 100,000
- Minimum years data retention: 10
- Low power consumption



The SH79F3218 embeds 32K flash program memory for program code. The flash program memory provides electrical erasure and programming and supports In-Circuit Programming (ICP) mode and Self-Sector Programming (SSP) mode. Every sector is 1024 bytes.

The SH79F323 also embeds 1024 bytes EEPROM-likea memory block for storing user data.Every sector is 256 bytes.It has 4 sectors.

Flash operation defined:

In-Circuit Programming (ICP):Through the Flash programmer to wipe the Flash memory, read and write operations.

Self-Sector Programming (SSP) mode:User Program code run in Program Memory to wipe the Flash memory, read and write operations.

Flash Memory Supports the Following Operations:

(1) Code Protection Control Mode

SH79F3218 code protection function provides a high-performance security measures for the user. Each partition has four modes are available.
Code protection mode 0: allow/forbid any programmer write/read operations (not including overall erasure).
Code protection mode 1: allow/forbid through MOVC instructions to read operation in other sectors, or through SSP mode to erased/write operation.
Code-protect control mode 2: Used to enable/disable the erase/write EEPROM operation through **SSP** Function.
Code-protect control mode 3: Customer password, write by customer, consists of 6 bytes. To enable the wanted protect mode, the user must use the Flash Programmer to set the corresponding protect bit.
The user must use the following two ways to complete code protection control mode Settings:
1. Flash programmer in ICP mode is set to corresponding protection bit to enter the protected mode.
2. The SSP mode does not support code protection control mode programming.

(2) Mass Erase

Regardless of the state of the code protection control mode, the overall erasure operation will erase all programs, code options, the code protection bit, but they will not erase EEPROM-like memory block.

The user must use the following way to complete the overall erasure:

Flash programmer in ICP mode send overall erasure instruction to run overall erasure.

The SSP mode does not support overall erasure mode.

(3) Sector Erase

Sector erasure operations will erase the content of selected sector. The user program (SSP) and Flash programmer can perform this operation.

For user programs to perform the operation, code protection mode 1 in the selected sector must be forbidden.

For Flash programmer to perform the operation,code protection mode 0 in the selected sector must be forbidden.

The user must use one of the following two ways to complete sector erasure:

1. Flash programmer in ICP mode send sector erasure instruction to run sector erasure.
2. Through the SSP function send sector erasure instruction to run sector erasure (see chapter SSP).

(4) EEPROM-like Memory Block Erasure

EEPROM-like memory block erasure operations will erase the content in EEPROM-like memory block.The user program (SSP) and Flash programmer can perform this operation.

The user must use one of the following two ways to complete EEPROM-like memory block erasure:

1. Flash programmer in ICP mode send EEPROM-like memory block erasure instruction to run EEPROM-like memory block erasure.
2. Through the SSP function send EEPROM-like memory block erasure instruction to run EEPROM-like memory block erasure (see chapter SSP).

(5) Write/Read Code

Write/read code operation can read or write code from flash memory block. The user program (SSP) and Flash programmer can perform this operation.

For user programs to perform the operation, code protection mode 1 in the selected sector must be forbidden. Regardless of the security bit Settings or not, the user program can read/write the sector which contains program itself.

For Flash programmer to perform the operation, code protection mode 0 in the selected sector must be forbidden.

The user must use one of the following two ways to complete write/read code:

1. Flash programmer in ICP mode send write/read code instruction to run write/read code.
2. Through the SSP function send write/read code instruction to run write/read code.

(6) Write/Read EEPROM-like Memory Block

EEPROM-like memory block operation can read or write data from EEPROM-like memory block. The user program (SSP) and Flash programmer can perform this operation.

The user must use one of the following two ways to complete write/read EEPROM-like memory block:

1. Flash programmer in ICP mode send write/read EEPROM-like memory block instruction to run write/read EEPROM-like memory block.
2. Through the SSP function send write/read EEPROM-like memory block instruction to run write/read EEPROM-like memory block.

Flash Memory Block Operation Summary

Operation	ICP	SSP
Code protection	support	non support
Sector erasure	support (no security bit)	support (no security bit)
Overall erasure	support	non support
EEPROM-like memory block erasure	support	support
Write/read code	Support (no security bit)	support (no security bit)
Read/write EEPROM-like memory block	support	support

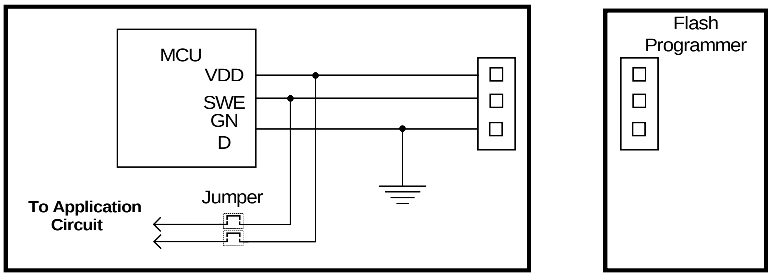
7.3.2 Flash Operation in ICP Mode

Singe Line Simulation Model:

ICP mode is performed without removing the micro-controller from the system. In ICP mode, the user system must be power-off, and the programmer can refresh the program memory through ICP programming interface. The ICP programming interface consists of 3 pins (V_{DD}, GND, SWE).

At first the one JTAG pins (SWE) are used to enter the programming mode. Only after the one pins are inputted the specified waveform, the CPU will enter the programming mode. For more detail description please refers to the FLASH Programmer's user guide.

In ICP mode, all the flash operations are completed by the programmer through 3-wire interface. Since the program signal is very sensitive, 3 jumpers are needed (V_{DD}, GND, SWE) to separate the program pins from the application circuit, as show in the following diagram.



When using ICP mode to do operations, the recommended steps are as following:

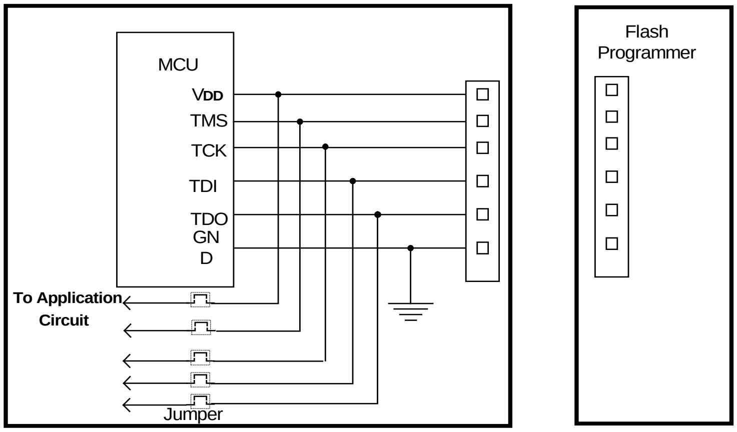
(1) The jumpers must be open to separate the programming pins from the application circuit before programming.
(2) Connect the programming interface with programmer and begin programming.
(3) Disconnect programmer interface and connect jumpers to recover application circuit after programming is complete.

Four Line Simulation Model:

ICP mode is performed without removing the micro-controller from the system. In ICP mode, the user system must be power-off, and the programmer can refresh the program memory through ICP programming interface. The ICP programming interface consists of 6 pins (V_{DD}, GND, TCK, TDI, TMS, TDO).

At first the four JTAG pins (TDO, TDI, TCK, TMS) are used to enter the programming mode. Only after the four pins are inputted the specified waveform, the CPU will enter the programming mode. For more detail description please refers to the FLASH Programmer's user guide.

In ICP mode, all the flash operations are completed by the programmer through 6-wire interface. Since the program signal is very sensitive, 6 jumpers are needed (V_{DD}, GND, TDO, TDI, TCK, TMS) to separate the program pins from the application circuit, as show in the following diagram.



When using ICP mode to do operations, the recommended steps are as following:

(1) The jumpers must be open to separate the programming pins from the application circuit before programming.
(2) Connect the programming interface with programmer and begin programming.
(3) Disconnect programmer interface and connect jumpers to recover application circuit after programming is complete.

7.4 SSP Function

7.4.1 SSP Register

(1) Memory Page Register for Programming

The register is used to select area code which will be erased or programmed, using IB_OFFSET register to show the address offset of bytes which is waiting for programming in the sector.

For program memory block, a sector is 1024 bytes, registers are defined as follows:

Table 7.4 Memory Page Register for Programming

F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-2	**XPAGE[7:2]**	Sector of the flash memory to be programmed, 0000 means sector 0, and so on
1-0	**XPAGE[1:0]**	High 2 Address of the flash memory sector to be erased/programmed

Table 7.5 Offset of Flash Memory for Programming

FBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_OFFSET	IB_OFFSET.7	IB_OFFSET.6	IB_OFFSET.5	IB_OFFSET.4	IB_OFFSET.3	IB_OFFSET.2	IB_OFFSET.1	IB_OFFSET.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_OFFSET[7:0]**	Low 8 Address of the flash memory sector to be programmed

XPAGE[1:0] and IB_OFFSET[7:0] are total 10 bit, they can be used to express the offset of 1024 bytes in a program memorysector.

For EEPROM-like memory block, a sector is 256 bytes, it has 8 sectors, registers are defined as follows:

Table 7.6 Memory Page Register for Programming/Erasing

F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-3	**XPAGE[7:3]**	Meaningless in erase/program sector
2-0	**XPAGE[2:0]**	**Sector select bit** 000: Sector 0 001: Sector 1 … 111: Sector 7

The address to the EEPROM-Like block can be achieved by “MOVC A, @A+DPTR” or “MOVC A, @A+PC”.

Note: *FAC bit in FLASHCON register should be set.*

Table 7.7 Offset of Flash Memory for Programming

FBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_OFFSET	IB_OFFSET.7	IB_OFFSET.6	IB_OFFSET.5	IB_OFFSET.4	IB_OFFSET.3	IB_OFFSET.2	IB_OFFSET.1	IB_OFFSET.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_OFFSET[7:0]**	Address of the flash memory to be erased/programmed

IB_OFFSET[7:0] is 8 bit, it can be used to express the offset of 256 bytes in a program memory sector.

(2) Data Register for Programming

Table 7.8 Data Register for Programming

FCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_DATA	IB_DATA.7	IB_DATA.6	IB_DATA.5	IB_DATA.4	IB_DATA.3	IB_DATA.2	IB_DATA.1	IB_DATA.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_DATA[7:0]**	Data to be programmed

(3) SSP Type select Register

Table 7.9 SSP Type select Register

F2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON1	IB_CON1.7	IB_CON1.6	IB_CON1.5	IB_CON1.4	IB_CON1.3	IB_CON1.2	IB_CON1.1	IB_CON1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_CON1[7:0]**	**SSP Type select** 0xE6: Sector Erase 0x6E: Sector Programming

(4) SSP Flow Control Register1

Table 7.10 SSP Flow Control Register1

F3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON2	-	-	-	-	IB_CON2.3	IB_CON2.2	IB_CON2.1	IB_CON2.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	**IB_CON2[3:0]**	Must be 05H, otherwise Flash Programming will terminate

Table 7.11 SSP Flow Control Register2

F4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON3	-	-	-	-	IB_CON3.3	IB_CON3.2	IB_CON3.1	IB_CON3.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	**IB_CON3[3:0]**	Must be 0AH, otherwise Flash Programming will terminate

Table 7.12 SSP Flow Control Register3

F5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON4	-	-	-	-	IB_CON4.3	IB_CON4.2	IB_CON4.1	IB_CON4.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

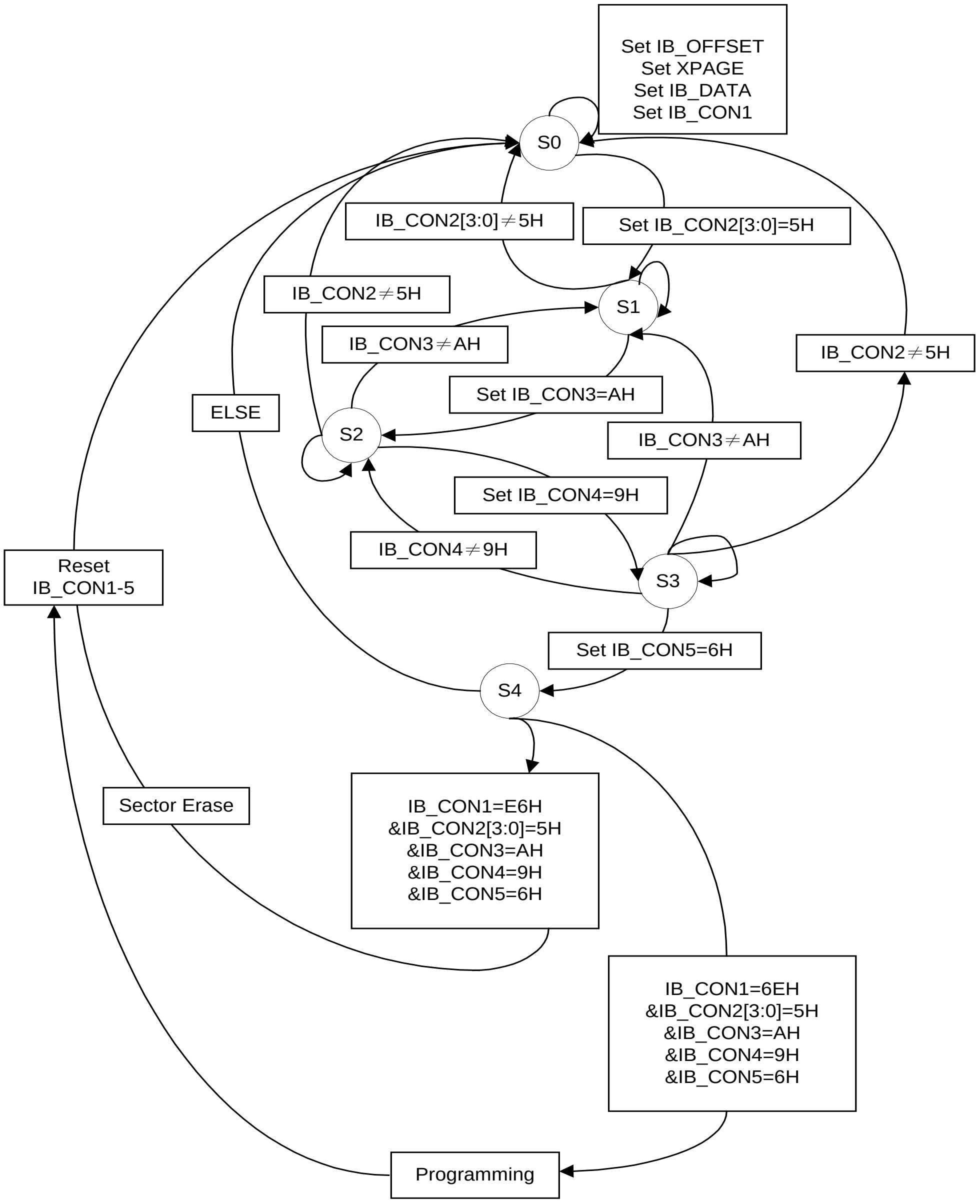
Bit Number	Bit Mnemonic	Description
3-0	**IB_CON4[3:0]**	Must be 09H, otherwise Flash Programming will terminate

Table 7.13 SSP Flow Control Register4

F6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON5	-	-	-	-	IB_CON5.3	IB_CON5.2	IB_CON5.1	IB_CON5.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	**IB_CON5[3:0]**	Must be 06H, otherwise Flash Programming will terminate

7.4.2 Flash Control Flow



7.4.3 SSP Programming Notice

To successfully complete SSP programming, the user's software must be set as the following the steps:

(1) For Code/Data Programming: ***Note:*** *must close Code-protect control mode 1 and Code-protect control mode 2.*

1. Disable interrupt;
2. Fill in the XPAGE, IB_OFFSET for the corresponding address;
3. Fill in IB_DATA, if programming is wanted;
4. Fill in IB_CON1-5 sequentially;
5. Add 4 nops for more stable operation;
6. Code/Data programming, CPU will be in IDLE mode;
7. Go to Step 2, if more data are to be programmed;
8. Clear XPAGE; enable interrupt if necessary.

(2) For Sector Erase: ***Note:*** *must close Code-protect control mode 1 and Code-protect control mode 2.*

1. Disable interrupt;
2. Fill in the XPAGE for the corresponding sector;
3. Fill in IB_CON1-5 sequentially;
4. Add 4 NOPs for more stable operation;
5. Sector Erase, CPU will be in IDLE mode;
6. Go to step 2, if more sectors are to be erased;
7. Clear XPAGE; enable interrupt if necessary.

(3) For Code Reading:

Just Use "MOVC A, @A+DPTR" or "MOVC A, @A+PC".

(4) For EEPROM-Like: ***Note:*** *The function is not controlled by code protect controlmode.*

The operation for EEPROM-Like is similar to the operation for Flash memory, the differences are:

1. FAC bit in FLASHCON register must be set before wipe, read or write EEPROM-Like.
2. EEPROM-Like sector is 256 bytes, rather than 1024 bytes.

Note: *FAC bit must be cleared when do not operate EEPROM-Like.*

7.4.4 Readable Random Code

Every chip is cured an 40-bit readable random code after production. Readable random code is 0 -0xffffffffff random value, and can not be erased, can be read by program or tools.

How to read random code: set FAC bit, Assigned to the DPTR as "0127BH - 127FH", clear A, then use "MOVC A, @A+DPTR" to read.

Note: *It is needed to clear FAC after reading readable random code, otherwise it will influence on the instructions execution of reading program ROM.*

FLASHCON register description is as follows:

Table 7.14 Flash Access Control Register

A7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FLASHCON	-	-	-	-	-	-	-	FAC
R/W	-	-	-	-	-	-	-	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-	-	-	0

Bit Number	Bit Mnemonic	Description
0	**FAC**	**FAC: Flash access control** 0: MOVC or SSP access Main Block 1: MOVC or SSP access EEPROM-like

7.5 System Clock and Oscillator

7.5.1 Features

- Six oscillator types: 32.768kHz crystal oscillator, crystal oscillator, ceramic oscillator and 16.6MHz /128kHz internal RC
- Two Oscillator pin (XTAL1, XTAL2)
- 16MHz Internal RC (T_A = +25°C, ±1%), (T_A = -40°C +85°C, ±2%)
- Built-in system clock prescaler

7.5.2 Clock Definition

SH79F3218 have several internal clocks defined as below:：

OSCCLK: the oscillator clock is selected from the four oscillator types (32.768kHz crystal oscillator, crystal oscillator, ceramic tor and internal RC 16.6M /128K oscillator from XTAL input). f_{OSC} is defined as the OSCCLK frequency. t_{OSC} is defined as the OSCCLK period.

OSC1CLK: the oscillator clock is select from the three oscillator types (32.768kHz crystal oscillator, crystal oscillator, ceramic oscillator and internal RC16.6M oscillator from XTALX input). f_{OSC} is defined as the OSC1CLK frequency. t_{OSC} is defined as the OSC1CLK period.

Note: *OSC1CLK does not exist when code option OP_OSC is not 0011，0110，1010 (32.768kHz/128kHz RC is not selected, Refer to* ***code option*** *section for details)*

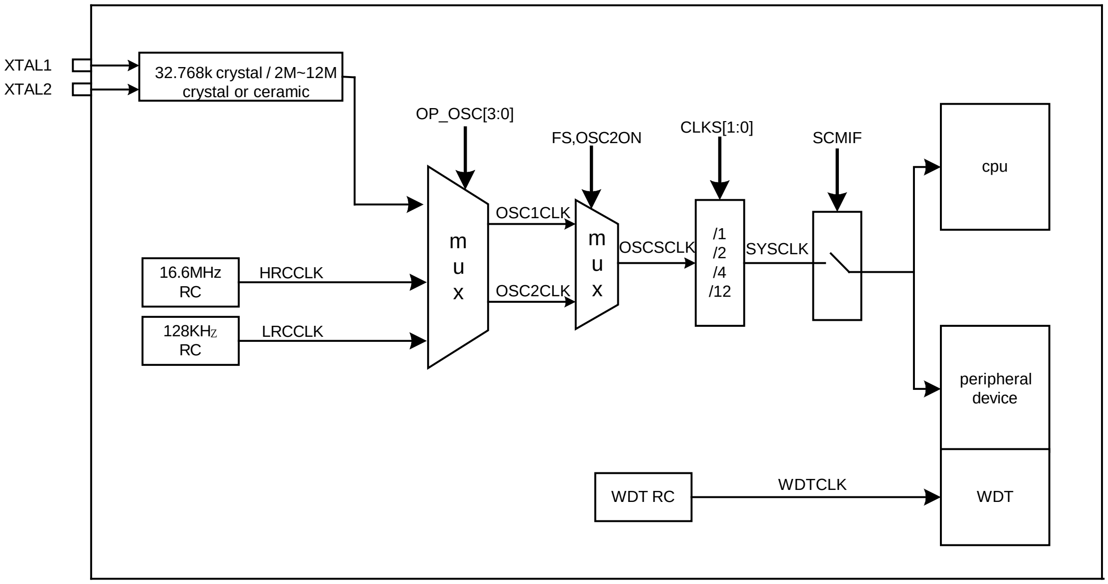
WDTCLK: the internal WDT RC clock. f_{WDT} is defined as the WDTCLK frequency. t_{WDT} is defined as the WDTCLK period.

OSCSCLK: the input clock of system clock frequency prescaler. It can be OSCCLK or OSC1CLK. f_{OSCS} is defined as the OSCSCLK frequency. t_{OSCS} is defined as the OSCSCLK period.

SYSCLK: stem clock, the output clock of system clock prescaler. It is the CPU instruction clock. f_{SYS} is defined as the SYSCLK frequency. t_{SYS} is defined as the SYSCLK period.

7.5.3 Description

SH79F3218 has three oscillator types: crystal oscillator(2MHz-12MHz)(32.768kHz),ceramic oscillator (2MHz-12MHz) and internal RC (16.6MHz, 128KHz), which is selected by code option OP_OSC (Refer to code option section for details). SH79F3218 has 2 oscillator pins (XTAL1, XTAL2), which can generate 1 or 2 clocks from 3 oscillator types. They also are selected by code option OP_OSC (Refer to code option section for details). The oscillator generates the basic clock pulse that provides the system clock to supply CPU and on-chip peripherals.



7.5.4 Register

Table 7.15 System Clock Control Register

B2H, Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	32k_SPDUP	CLKS1	CLKS0	SCMIF	HFON	FS	-	-
R/W	R/W	R/W	R/W	R	R/W	R/W	-	-
Reset Value (POR/WDT/LVR/PIN)	1	1	1	0	0	0	-	-

Bit Number	Bit Mnemonic	Description
7	**32k_SPDUP**	**32.768kHz oscillator speed up mode control bit** 0: 32.768kHz oscillator normal mode, cleared by software. 1: 32.768kHz oscillator speed up mode, set by hardware or software. This control bit is set by hardware automatically in all kinds of RESET such as Power on reset, watch dog reset etc. to speed up the 32.768kHz Oscillator oscillating, shorten the 32.768kHz oscillator start-oscillating time. And this bit also can be set or cleared by software if necessary. Such as set before entering Power-down mode and cleared when Power-down mode wakes up. It should be noticed that turning off 32.768kHz oscillator speed up (clear this bit) could reduce the system power consumption. Only when code option OP_OSC is 1010, this bit is valid. (32.768kHz oscillator is selected, Refer to **code option** section for details)
6-5	**CLKS[1:0]**	**SYSCLK Prescaler Register** 00: $f_{SYS} = f_{OSCS}$ 01: $f_{SYS} = f_{OSCS}/2$ 10: $f_{SYS} = f_{OSCS}/4$ 11: $f_{SYS} = f_{OSCS}/12$ If 32.768kHz oscillator is selected as OSCSCLK, these control bits is invalid.
3	**HFON**	**OSC1CLK On-Off control Register** 0: turn off OSC1CLK 1: turn on OSC1CLK Only when code option OP_OSC is 0011, 0110, 1010 this bit is valid. (32.768kHz oscillator/128kHz internal RC is selected, refer to **code option** section for details)。
2	**FS**	**Frequency Select Register** 0: 32.768kHz/128kHz is selected as OSCSCLK. 1: OSC1CLK is selected as OSCSCLK. Only when code option OP_OSC is 0011, 0110, 1010 this bit is valid. (32.768kHz oscillator/128kHz internal RC is selected, refer to **code option** section)

Note:

(1) If code option OP_OSC is 0000, OSC1CLK is Internal 16.6M RC, if code option OP_OSC is 0110, 1010, OSC1CLK is osclator from XTALX input.

(2) HFON and FS is valid only when code option OP_OSC is 0011, 0110, 1010.

(3) When OSC1CLK is used as OSCSCLK (that is HFON = 1 and FS = 1), HFON is can't be cleared by software.

(4) When OSCSCLK changed from 32.768kHz/128kHz to OSC1CLK, if OSC1CLK is off, the setting must be done as the following steps:

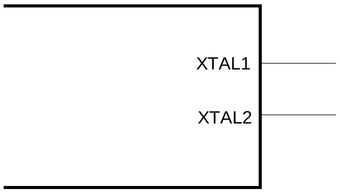
a. Set HFON = 1 to turn on the OSC1CLK
b. Wait at least Oscillator Warm-up timer (Refer to Warm-up Timer section for details)
c. Set FS = 1 to select OSC1CLK as OSCSCLK.

(5). When OSCSCLK changed from OSC1CLK to 32.768kHz/128kHz, the setting must be done as the following steps:

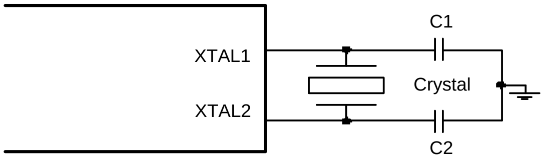
a. Clear FS to select 32.768kHZ/128k as OSCSCLK.
b. Add one NOP
c. Clear HFON(reduce power consumption)
d. Add four NOPs.

7.5.5 Oscillator Type

(1) OP_OSC = 0000, 0011: nternal RC 16.6M/128K, XTAL shared with IO.



(2) OP_OSC = 0110, 1010, 1110: 32.768kHz Crystal Oscillator at XTAL, 2M - 16M Crystal/Ceramic oscillator from XTAL input



7.5.6 Capacitor Selection for Oscillator

Ceramic Oscillator		
Frequency	Frequency	Frequency
3.58MHz	-	-
4MHz	-	-

Crystal Oscillator		
Frequency	Frequency	Frequency
4MHz	8 - 15pF	8 - 15pF
12MHz	8 - 15pF	8 - 15pF
32.768KHz	10 - 12pF	10 - 12pF

Note :

(1) Capacitor values are used for design guidance only!

*(2) These capacitors were tested with the crystals listed above for basic start-up and operation. They are **not optimized.***

(3) Be careful for the stray capacitance on PCB board, the user should test the performance of the oscillator over the expected V_{DD} and the temperature range for the application.

(4) Before selecting crystal/ceramic, the user should consult the crystal/ceramic manufacturer for appropriate value of external component to get best performance, visit http://www.sinowealth.com for more recommended manufactures.

7.6 System Clock Monitor (SCM)

In order to enhance the system reliability, SH79F3218 contains a system clock monitor (SCM) module. If the system clock breaks down(for example the external oscillator stops oscillating), the built-in SCM will switch the OSCCLK to the internal RC 2M clock, and set system clock monitor bit (SCMIF) to 1. And the SCM interrupt will be generated when EA and ESCM is enabled. If the external oscillator comes back, SCM will switch the OSCCLK back to the external oscillator and clears the SCMIF automatically.

Select SCM clock by set up SCMCON, if the built-in SCM detect the system clock breaks down, that will switch the OSCCLK to the internal SCM clock.

The SCM function is valid when using external clocl only.

Notes:

The SCMIF is read only register; it can be clear to 0 or set to 1 by hardware only.

If SCMIF is cleared, the SCM switches the system clock to the state before system clock breaks down automatically.

If Internal RC is selected as OSCSCLK by code option (Refer to **code option** section for detail), the SCM can not work.

Table 7.16 System Clock Control Register

B2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	-	-	-	SCMIF	-	-	-	-
R/W	-	-	-	R	-	-	-	-
Reset Value (POR/WDT/LVR/PIN	-	-	-	0	-	-	-	-

Bit Number	Bit Mnemonic	Description
4	SCMIF	**System Clock Monitor flag bit** 0: Clear by hardware to indicate system clock is normal 1: Set by hardware to indicate system clock fails

7.7 I/O Port

7.7.1 Features

- 22 bi-directional I/O ports
- Share with alternative functions

The SH79F3218 has 22 bi-directional I/O ports. The PORT data is put in Px register. The PORT control register (PxCRy) controls the PORT as input or output. Each I/O port has an internal pull-high resistor, which is controlled by PxPCRy when the PORT is used as input (x = 0-3, y = 0-7).

For SH79F3218, some I/O pins can share with alternative functions. There exists a priority rule in CPU to avoid these functions conflicts when all the functions are enabled. (Refer to **Port Share** Section for details)

The unused I/O needs to set the output (keep the low level) or input on the pull (keep the high level), avoiding the leakage and interference caused by the pin floating.

P3.3, P3.4, P3.6, P0.0 need to set. (PxPCRy, x = 0-4, y = 0-7) (x.y, x = 0-4, y = 0-7)

7.7.2 Register

Table 7.17 Port Control Register

E1H - E5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0CR (E1H, Bank0)	P0CR.7	P0CR.6	P0CR.5	P0CR.4	P0CR.3	P0CR.2	P0CR.1	P0CR.0
P1CR (E2H, Bank0)	P1CR.7	*	*	*	*	P1CR.2	P1CR.1	P1CR.0
P2CR (E3H, Bank0)	P2CR.7	P2CR.6	P2CR.5	P2CR.4	P2CR.3	P2CR.2	P2CR.1	P2CR.0
P3CR (E4H, Bank0)	*	P3CR.6	*	P3CR.4	P3CR3	P3CR.2	P3CR.1	P3CR.0
P4CR (91H, Bank1)	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**PxCRy** **x = 0-5, y = 0-7**	**Port input/output control Register** 0: input mode 1: output mode

Note: *the register is marked with * bit, need to set 1.*

Table 7.18 Port Pull up Resistor Control Register

E9H - ECH, 92H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0PCR (E9H, Bank0)	P0PCR.7	P0PCR.6	P0PCR.5	P0PCR.4	P0PCR.3	P0PCR.2	P0PCR.1	P0PCR.0
P1PCR (EAH, Bank0)	P1PCR.7	*	*	*	*	P1PCR.2	P1PCR.1	P1PCR.0
P2PCR (EBH, Bank0)	P2PCR.7	P2PCR.6	P2PCR.5	P2PCR.4	P2PCR.3	P2PCR.2	P2PCR.1	P2PCR.0
P3PCR (ECH, Bank0)	*	P3PCR.6	*	P3PCR.4	P3PCR.3	P3PCR.2	P3PCR.1	P3PCR.0
P4PCR (92H, Bank1)	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**PxPCRy** **x = 0-5, y = 0-7**	**Input Port internal pull-high resistor enable/disable control** 0: internal pull-high resistor disabled 1: internal pull-high resistor enabled

Note: *the register is marked with * bit, need to clear 0.*

Table 7.19 Port Data Register

80H - B0H, 90H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0 (80H, Bank0)	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P1 (90H, Bank0)	P1.7	*	*	*	*	P1.2	P1.1	P1.0
P2 (A0H, Bank0)	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P3 (B0H, Bank0)	*	P3.6	*	P3.4	P3.3	P3.2	P3.1	P3.0
P4 (90H, Bank1)	*	*	*	*	*	*	*	*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**Px.y x = 0-5, y = 0-7**	**Port Data Register**

***Note:** the register is marked with * bit, need to clear 0.*

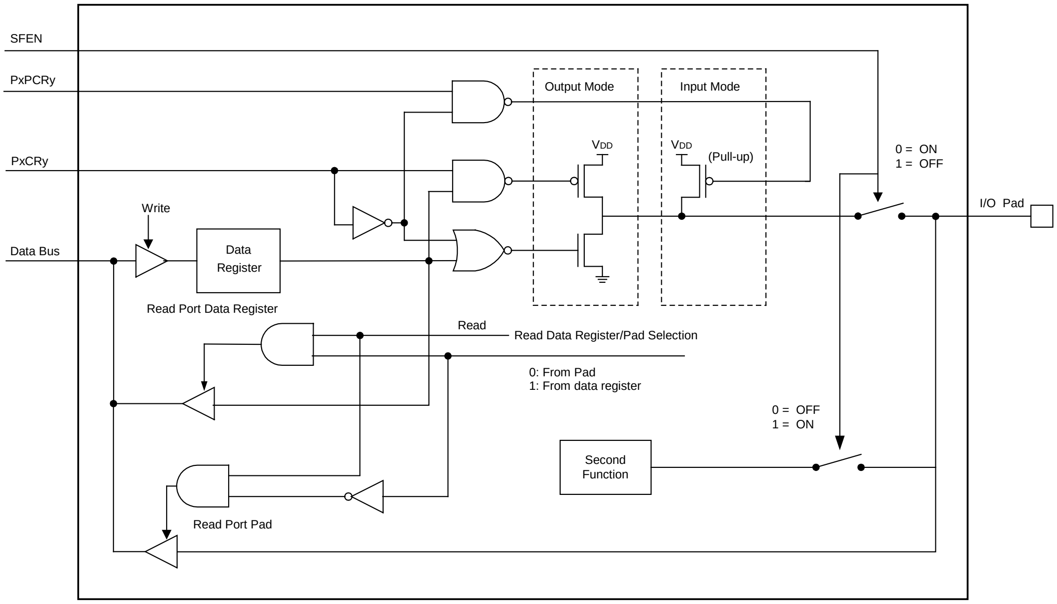
Table 7.20 Port mode select Register

EFH，Bank0	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位
P1OS	-	-	-	-	P1OSCR.1	P1OSCR.0	P1OS.1	P1OS.0
读/写	-	-	-	-	读/写	读/写	读/写	读/写
复位值 (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

位编号	位符号	说明
3-2	**P1OSCR.x x = 1-0**	**Internal pull-high resistor disabled of N-channel open drain mode** 0: Disable internal pull-high resistor 1: Enable internal pull-high resistor (10K)
1-0	**P1OS.x x = 1-0**	**Port0 output mode select** 0: output mode is N-channel open drain 1: Port output mode is CMOS Port

***Note:** P0.7, P1.0 art configured as N-channel open drain I/O, but voltage provided for this pin can't exceed V_{DD} + 0.3V. (P1OS.0 set P0.7, P1OS.1 set P1.0)*

7.7.3 Port Diagram



Note:

(1) The input source of reading input port operation is from the input pin directly.

(2) The input source of reading output port operation has two paths, one is from the port data Register, and the other is from the output pin directly.

(3) The read Instruction distinguishes which path is selected: The read-modify-write instruction is for the reading of the data register in output mode, and the other instructions are for reading of the output pin directly.

7.7.4 Port Share

The 22 bi-directional I/O ports can also share second or third special function. But the share priority should obey the **Outer Most Inner Lest** rule:

The out most pin function in **Pin Configuration** has the highest priority, and the inner most pin function has the lowest priority. This means when one pin is occupied by a higher priority function (if enabled), it cannot be used as the lower priority functional pin, even the lower priority function is also enabled. Only until the higher priority function is closed by hardware or software,can the corresponding pin be released for the lower priority function use. Also the function that need pull up resister is also controlled by the same rule.

When port share function is enabled, the user can modify PxCR, PxPCR (x = 0-4), but these operations will have no effect on the port status until the second function was disabled.When port share function is enabled, any read or write operation to port will only affect the data register.The value of the port pin kepps unchanged until the second function was disabled.

PORT0:

- SEG14 - 15: LED Segment14 - 15 (P0.5 - P0.6)
- COM1: LED COM1 (P0.7)
- SDA: SDA pin (P0.7)
- RXD: EUART data input (P0.7)
- INT30: External inturrupt 30 (P0.7)
- BUZ1, 2: Buzzer output 1, 2 (P0.3, P0.6)

Table 7.21 PORT0 Share Table

Pin No.	Priority	Function	Enable bit
10	1	P0.1	I/O
11	1	P0.2	I/O
12	1	BUZ1	Set **BZEN1** bit in **BUZCON** Register
	2	P0.3	Above condition is not met
13	1	P0.4	I/O
14	1	SEG15	Set **SEG15** bit in **SEG02** register
	3	P0.5	Above condition is not met
15	1	SEG14	Set **SEG14** bit in **SEG02** register
	2	BUZ2	Set **BZEN2** bit in **BUZCON** Register
	3	P0.6	Above condition is not met
16	1	COM1	Set **COM1** bit in **LEDCOM** register
	2	SDA	When **ENTWI = 1**, do operations on **TWIDAT** register
	3	RXD	Read to **SBUF** Register
	4	INT30	Set **EX3** bit in **IEN1** register and set **EXS30** bit in **EXF1** register
	5	P0.7	Above condition is not met

Note: *P0.7, P1.0 art open drain port when P1OS is 00H.*

PORT1:

- INT31: External inturrupt 31 (P1.0)
- TXD: EUART data output (P1.1)
- SCK: SCK pin (P1.1)
- COM2 - 4: LED COM2 - 4 (P1.0 - P1.2)
- INT40, INT45, INT47: External inturrupt 40, 45, 47 (P1.1 - P1.2, P1.7)
- TCH0: Touch Key channel 0 (P1.7)
- SEG7, SEG13: LED Segment 7,13 (P1.2, P1.7)

Table 7.22 PORT1 Share Table

Pin No.	Priority	Function	Enable bit
17	1	COM2	Set **COM2** bit in **LEDCOM** register
	2	SCK	When **ENTWI = 1**, do operations on **TWIDAT** register
	3	TXD	Write to **SBUF** Register
	4	INT31	Set **EX3** bit in **IEN1** register and set **EXS31** bit in **EXF1** register
	5	P1.0	Above condition is not met
18	1	COM3	Set **COM3** bit in **LEDCOM** register
	2	INT47	Set **EX4** bit in **IEN1** register and set **EXS47** bit in **IENC** register
	3	P1.1	Above condition is not met
19	1	COM4	Set **COM4** bit in **LEDCOM** register
	2	SEG13	Set **SEG13** bit in **SEG02** register
	3	INT40	Set **EX4** bit in **IEN1** register and set **EXS40** bit in **IENC** register
	4	P1.2	Above condition is not met
20	1	TCH0	Set **P2SS.0** bit in **P2SS** register
	2	SEG7	Set **SEG7** bit in **SEG01** register
	3	INT45	Set **EX4** bit in **IEN1** register and set **EXS45** bit in **IENC** register
	4	P1.7	Above condition is not met

Note: *P0.7，P1.0 art open drain port when P1OS is 00H.*

PORT2:

- INT46: External inturrupt 46 (P2.2)
- PWM1 - PWM2: PWM1/2 Output (P2.0 - P2.1)
- SEG0 - 6: LED Segment 0-6 (P2.0 - P2.7)
- TCH1 - TCH7: Touch Key channel 1 - 7 (P2.0 - P2.7)
- C: Touch key external CAP pin (P2.7)

Table 7.23 PORT2 Share Table

Pin No.	Priority	Function	Enable bit
21	1	TCH1	Set **P2SS.1** bit in **P2SS** register
	2	SEG6	Set **SEG6** bit in **SEG01** register
	3	PWM2	Set **PWM2EN** bit in **PWM2CON** register
	4	P2.0	Above condition is not met
22	1	TCH2	Set **P2SS.2** bit in **P2SS** register
	2	SEG5	Set **SEG5** bit in **SEG01** register
	3	PWM1	Set **PWM1EN** bit in **PWM1CON** register
	4	P2.1	Above condition is not met
23	1	TCH3	Set **P2SS.3** bit in **P2SS** register
	2	SEG4	Set **SEG4** bit in **SEG01** register
	3	INT46	Set **EX4** bit in **IEN1** register and set **EXS46** bit in **IENC** register
	4	P2.2	Above condition is not met
24 - 3	1	TCH4 - TCH7	Set **P2SS.4-7** bit in **P2SS** register
	2	SEG0 - SEG3	Set **SEG0-3** bit in **SEG01** register
	3	P2.3 - 2.6	Above condition is not met
4	1	C	Set **TKCON** bit in **TKCON1** register
	2	COM7	Set **COM7** bit in **LEDCOM** register
	3	P2.7	Above condition is not met

PORT3:

- T3: Timer3 external input (P3.1)
- T4: Timer4 external input (P3.0)
- INT2: External inturrupt2 (P3.0)
- SEG11, 12: LED Segment 11-12 (P3.0, P3.1)
- COM7: LED COM7 (P3.1)
- SEW: One line Debug interface (P3.2)

Table 7.24 PORT3 Share Table

Pin No.	Priority	Function	Enable bit
5	1	XTAL1	Code option
	2	COM6	Set **COM6** bit in **LEDCOM** register
	3	SEG12	Set **SEG12** bit in **SEG02** register
	4	T4	Set **TR4** bit and **T4CLKS** bit in **T4CON** register (Auto Pull up) or clear **T4CLKS** bit and set **TC4** bit or set **TR4** bit in Mode2
	5	P3.0	Above condition is not met
6	1	XTAL2	Code option
	2	COM5	Set **COM5** bit in **LEDCOM** register
	3	SEG11	Set **SEG11** bit in **SEG02** register
	4	T3	Set **TR3** bit in **T3CON** register and **T3CLKS[1:0] = 01** (Auto Pull up)
	5	P3.1	Above condition is not met
7	1	SWE	One line Debug interface
	2	INT2	Set **EX2** bit in **IEN1** register, P3.2 as input port
	3	P3.2	Above condition is not met

7.8 Timer

7.8.1 Features

- The SH79F3218 has five timers (Timer0, 1, 2, 3, 4)
- Timer2 is a 16-bit auto-reload timer
- Timer3 is a 16-bit auto-reload timer and can operate even in Power-Down mode
- Timer4 is a 16-bit auto-reload timer, two data register: TH4 & TL4 can be used as a 16-bit register to access

7.8.2 Timer0/1

Each timer is implemented as a 16-bit register accessed as two cascaded Timer x/Counter x Data Registers: THx & TLx (x = 0, 1). They are controlled by the register TCON and TMOD. The Timer 0 & Timer 1 interrupts can be enabled by setting the ET0 & ET1 bit in the IEN0 register (Refer to **Interrupt** Section for details).

Timer0 & Timer1 Mode

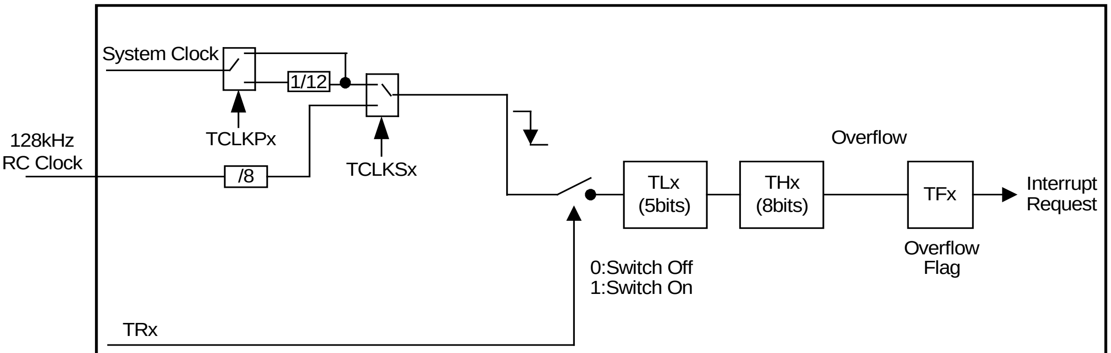
Both timers operate in one of four primary modes selected by the Mode Select bits Mx1-Mx0 (x = 0, 1) in the Counter/Timer Mode register (TMOD).

Mode0: 13-bit Counter/Timer

Timer x operate as 13-bit counter/timers in Mode 0. The THx register holds the high eight bits of the 13-bit counter/timer, TLx holds the five low bits TLx.4- TLx.0. The three upper bits(TLx.7- TLx.5) of TLx are indeterminate and should be ignored when reading. As the 13-bit timer register increments and overflows, the timer overflow flag TFx is set and an interrupt will occur if Timer interrupts is enabled. The C/$\overline{\text{Tx}}$ bit selects the counter/timer's clock source.

When as Timer, system clock or 1/8 of 128K clock can be selected as Timer x (x = 0, 1) clock source by configuring TCLKSx (x = 0, 1) in TCON1 Register.

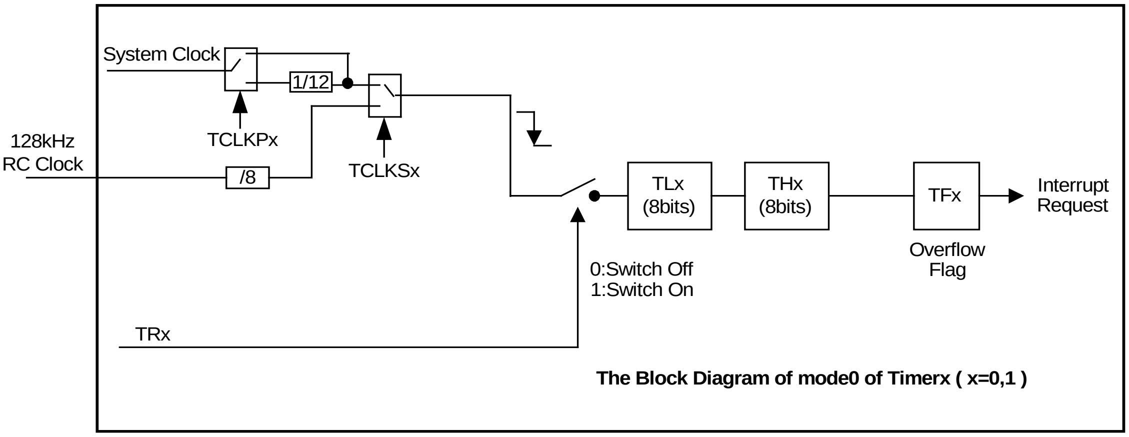
System clock or 1/12 of system clock can be selected as Timer x (x = 0, 1) clock source by configuring TCLKPx (x = 0, 1) in TCON1 Register.



The Block Diagram of mode0 of Timerx (x=0,1)

Mode1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.



The Block Diagram of mode0 of Timerx (x=0,1)

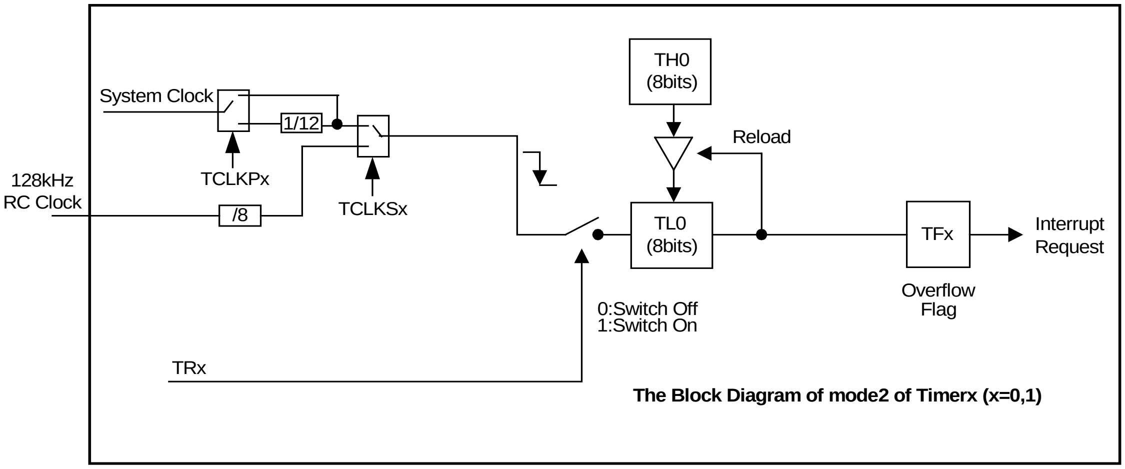
Mode2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TLx holds the count and THx holds the reload value. When the counter in TLx overflows from 0xFF to THx, the timer overflow flag TFx is set and the counter in TLx is reloaded from THx. If Timer 0 interrupts are enabled, an interrupt will occur when the TFx flag is set. The reload value in TH0 is not changed. TLx 0 must be initialized to the desired value before enabling the timer for the first count to be correct.

Except the Auto-Reload function, both counter/timers are enabled and configured in Mode 2 is the same as in Mode 0 & Mode 1.

When as Timer, system clock or 1/8 of 128K clock can be selected as Timer x (x = 0, 1) clock source by configuring TCLKSx (x = 0, 1) in TCON1 Register.

System clock or 1/12 of system clock can be selected as Timer x (x = 0, 1) clock source by configuring TCLKPx (x = 0, 1) in TCON1 Register.



The Block Diagram of mode2 of Timerx (x=0,1)

Mode3: Two 8-bit Counter/Timers (Timer 0 Only)

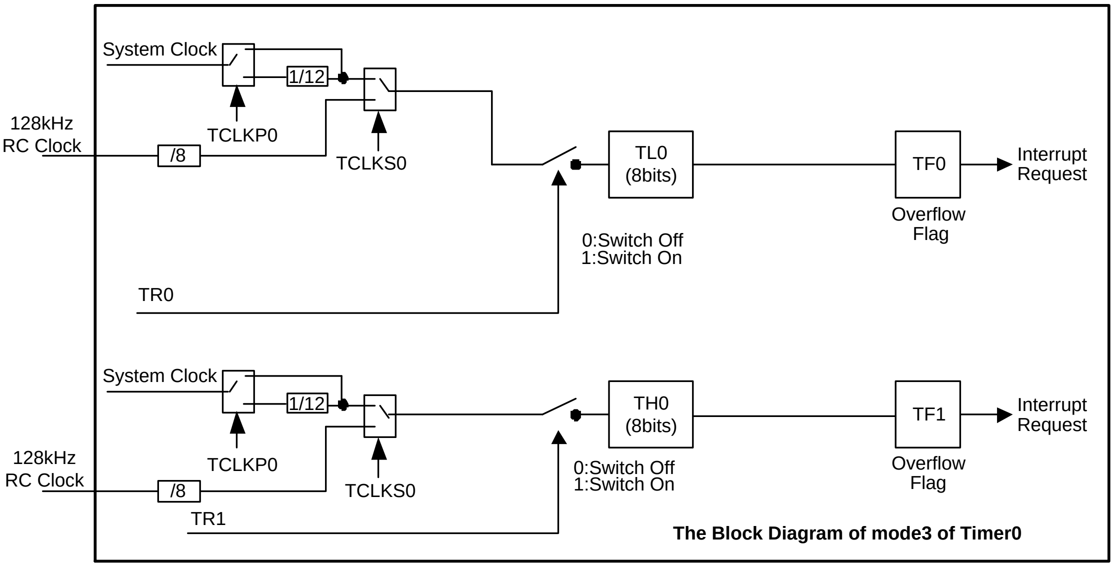
In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0 and TF0. TL0 can use either the system clock or 1/8 of 128K clock or an external input signal as its time base.

The TH0 is restricted to a timer function sourced by the system clock. TH0 is enabled using the Timer 1 control bit TR1. THx sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

When timer 0 is operating in Mode 3, timer 1 can be operated in modes 0, 1 or 2, but it cannot set the TF1 flag and generate an interrupt. The TH1 and TL1 register is restricted to a timer function sourced by the system clock. Timer 1 run control is handled through its mode settings, because TR1 is used by Time 0. When the timer 1 is in mode 0, 1, or 2, timer 1 is enable. When the timer 1 is in mode 3, timer 1 is disable.

When as Timer, system clock or 1/8 of 128K clock can be selected as Timer x (x = 0, 1) clock source by configuring TCLKSx (x = 0, 1) in TCON1 Register.

System clock or 1/12 of system clock can be selected as Timer0 clock source by configuring TCLKP0 in TCON1 Register.



The Block Diagram of mode3 of Timer0

Registers

Note: *Timer 0/1 register is not defined by the blank space must be clear 0.*

Table 7.25 Timer/Counter x Control register (x = 0, 1)

88H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	-	-	-	-
R/W	R/W	R/W	R/W	R/W				
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0				

Bit Number	Bit Mnemonic	Description
7, 5	**TFx** **x = 0, 1**	**Timer x overflow flag** 0: Timer x no overflow, can be cleared by software 1: Timer x overflow, set by hardware; set by software will cause a timer interrupt
6, 4	**TRx** **x = 0, 1**	**Timer x start, stop control bits** 0: Stop timer x 1: Start timer x

Table 7.26 Timer/Counter x Mode Register (x = 0, 1)

89H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMOD	-	-	M11	M10	-	-	M01	M00
R/W	-	-	R/W	R/W	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	0	0	-	-	0	0

Bit Number	Bit Mnemonic	Description
5-4 **1-0**	**Mx[1:0]** **x = 0, 1**	**Timer x Timer mode selected bits** 00: Mode 0, 13-bit up counter/timer, bit7-5 of TLx is ignored. 01: Mode 1, 16-bit up counter/timer 10: Mode 2, 8-bit auto-reload up counter/timer 11: Mode 3 (only for Timer0), two 8-bit up timer

Note: *Timer 0/1 register is not defined by the blank space must be clear 0.*

Table 7.27 Timer x Counter Register (x = 0, 1)

8AH-8DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL0 (8AH)	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
TH0 (8BH)	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
TL1 (8CH)	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
TH1 (8DH)	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**TLx.y, THx.y** **x = 0-1, y = 0-7**	**Timer x Low & High byte counter**

Table 7.28 Timer x Control Register 1 (x = 0, 1)

8FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON1	-	TCLK_S1	TCLK_S0	-	TCLKP1	TCLKP0	-	-
R/W	-	R/W	R/W	-	R/W	R/W	-	-
Reset Value (POR/WDT/LVR/PIN)	-	0	0	-	0	0	-	-

Bit Number	Bit Mnemonic	Description
6-5	**TCLK_Sx** **x = 0, 1**	**Timer x Clock Source Prescale bits** 0: Select system clock as Timer x Clock Source 1: Select 1/8 of 128K RC clock as Timer x Clock Source
3-2	**TCLKPx** **x = 0, 1**	**Timer x Clock Source Prescale bits** 0: Select 1/12 of system clock as Timer x Clock Source 1: Select system clock as Timer x Clock Source

7.8.3 Timer2

The Timer2 is implemented as a 16-bit register accessed as two cascaded data registers: TH2 and TL2. It is controlled by the register T2CON and T2MOD. The Timer2 interrupt can be enabled by setting the ET2 bit in the IEN0 register. (Refer to Interrupt Section for details)

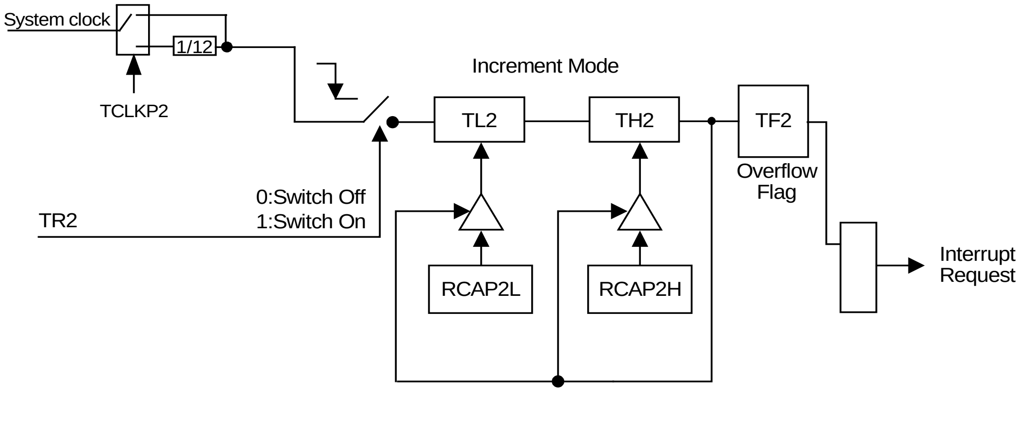
Setting TR2 allows Timer2/Counter2 Data Register to increment by the selected input.

Timer2 Modes

Mode0: 16 bit auto-reload Timer

Timer 2 can be programmed to count up when configured in its 16-bit auto-reload mode.

Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.



The Block Diagram of Auto Relode Mode (Mode 1)of Timer2 (DCEN=0)

Note:

(1) TF2 is set as 1 by hardware while event occurs. But they can also be set by software at any time. Only the software and the hardware reset will be able to clear TF2 to 0.

(2) When EA = 1 & ET2 = 1, setting TF2 as 1 will cause a timer2 interrupt.

Registers

Note: *Timer 2 register is not defined by the blank space must be clear 0.*

Table 7.29 Timer2 Control Register

C8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	TF2	-	-	-	-	TR2	-	-
R/W	R/W	-	-	-	-	R/W	-	-
Reset Value (POR/WDT/LVR/PIN)	0	-	-	-	-	0	-	-

Bit Number	Bit Mnemonic	Description
7	**TF2**	**Timer2 overflow flag bit** 0: No overflow 1: Overflow (Set by hardware if RCLK = 0 & TCLK = 0)
2	**TR2**	**Timer2 start/stop control bit** 0: Stop Timer2 1: Start Timer2

Table 7.30 Timer2 Mode Control Register

C9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2MOD	TCLKP2	-	-	-	-	-	-	-
R/W	R/W	-	-	-	-	-	-	-
Reset Value (POR/WDT/LVR/PIN)	0	-	-	-	-	-	-	-

Bit Number	Bit Mnemonic	Description
7	**TCLKP2**	**Timer 2 Clock Source Prescale bits** 0: Select system clock as Timer2 Clock Source 1: Select 1/12 of system clock as Timer2 Clock Source

Table 7.31 Timer2 Reload/Capture Registers

CAH-CDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCAP2L	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
RCAP2H	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
TL2	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
TH2	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**RCAP2L.x**	Timer2 Reload/Capturer Data, x = 0-7
	RCAP2H.x	
7-0	**TL2.x**	Timer2 Low & High byte counter, x = 0-7
	TH2.x	

7.8.4 Timer3

Timer3 is a 16-bit auto-reload timer. It is accessed as two cascaded Data Registers: TH3 and TL3. It is controlled by the T3CON register. The Timer3 interrupt can be enabled by setting ET3 bit in IEN1 register (Refer to **Interrupt** Section for details).

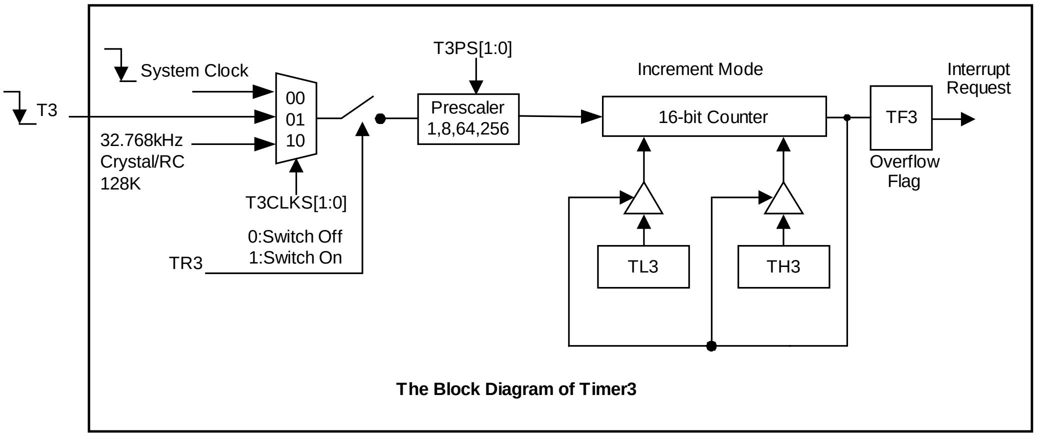
Timer3 has only one operating mode: 16-bit Counter/Timer with auto-reload. Timer3 also supports the following features: selectable pre-scaler setting and Operation during CPU Power-Down mode.

Timer3 consists of a 16-bit counter/reload register (TH3, TL3). When writing to TH3 and TL3, they are used as timer load register. When reading from TH3 and TL3, they are used as timer counter register. Setting the TR3 bit enables Timer 3 to count up. The Timer will overflow from 0xFFFF to 0x0000 and set the TF3 bit. This overflow also causes the 16-bit value written in timer load register to be reloaded into the timer counter register. Writing to TH3 also can cause the 16-bit value written in timer load register to be reloaded into the timer counter register.

Read or write operation to TH3 and TL3 should follow these steps:

Write operation: Low bits first, High bits followed.

Read operation: High bits first, Low bits followed.



The Block Diagram of Timer3

Timer3 can operate even in Power-Down mode.

When OP_OSC[3:0] is 0000,1110 T3CLKS[1:0] can be selected as 00, 01 (refer to “Code Option” section for details). When OP_OSC[3:0] is 0011,0110, 1010,T3CLKS[1:0] can be selected as 00, 01, 10.

If T3CLKS[1:0] is 00, Timer3 can’t work in Power-Down mode. If T3CLKS[1:0] is 01, when T3 port input external clock, Timer3 can work in CPU normal operating or Power Down mode (entering Power Down mode when system clock is high frequency). If T3CLKS[1:0] is 10 and OP_OSC[3:0] is 0011,0110,1010, Timer3 can work in CPU normal operating or Power Down mode. If T3CLKS[1:0] is 10 and OP_OSC[3:0] is 0000,1110, Timer3 can’t work. It can be described in the following table:

OP_OSC[3:0]	T3CLKS[1:0]	Can work in normal mode	Can work in Power Down mode
0000 1110	00	YES	NO
	01	YES	YES
	10	NO	NO
0011 0110 1010	00	YES	NO
	01	YES	YES
	10	YES	YES

Note:

(1) When TH3 and TL3 read or written, must make sure TR3 = 0.

(2) When T3 is selected as Timer3 clock source and TR3 is set 0 to 1, the first T3 down edge will be ignored.

Registers

Table 7.32 Timer3 Control Register

C0H, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T3CON	TF3	-	T3PS.1	T3PS.0	-	TR3	T3CLKS.1	T3CLKS.0
R/W	R/W	-	R/W	R/W	-	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	-	0	0	-	0	0	0

Bit Number	Bit Mnemonic	Description
7	**TF3**	**Timer3 overflow flag bit** 0: No overflow (cleared by hardware) 1: Overflow (Set by hardware)
5-4	**T3PS[1:0]**	**Timer3 input clock Prescaler Select bits** 00: 1/1 01: 1/8 10: 1/64 11: 1/256
2	**TR3**	**Timer3 start/stop control bit** 0: Stop Timer3 1: Start Timer3
1-0	**T3CLKS[1:0]**	**Timer3 Clock Source select bits** 00: System clock, T3 pin is used as I/O port 01: External clock from pin T3, auto pull-up 10: 32.768kHz from external Crystal 11: reserved

Table 7.33 Timer3 Reload/Counter Data Registers

C2H-C3H, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL3 (C2H)	TL3.7	TL3.6	TL3.5	TL3.4	TL3.3	TL3.2	TL3.1	TL3.0
TH3 (C3H)	TH3.7	TH3.6	TH3.5	TH3.4	TH3.3	TH3.2	TH3.1	TH3.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**TL3.x**	Timer3 Low & High byte counter, x = 0 - 7
	TH3.x	

Table 7.34 Timer3 Reload/Counter Data Mode Registers

C1H, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SWTHL	-	-	-	-	-	-	-	TH3LCON
R/W		-	-	-	-	-	-	R/W
Reset Value (POR/WDT/LVR/PIN)		-	-	-	-	-	-	0

Bit Number	Bit Mnemonic	Description
0	**T3HLCON**	0: when read TH3, TL3, return T3 count data 1: when read TH3, TL3, return T3 reload register data

7.8.5 Timer4

Timer4 is a 16-bit auto-reload timer. It is accessed as two cascaded data registers: TH4 and TL4. It is controlled by the T4CON register. The Timer 4 interrupt can be enabled by setting ET4 bit in IEN1 register (Refer to **interrupt** Section for details).

When writing to TH4 and TL4, they are used as timer load register. When reading from TH4 and TL4, they are used as timer counter register. Setting the TR4 bit enables Timer 4 to count up. The timer will overflow from 0xFFFF to 0x0000 and set the TF4 bit. This overflow also causes the 16-bit value written in timer load register to be reloaded into the timer counter register. Writing to TH4 also can cause the 16-bit value written in timer load register to be reloaded into the timer counter register.

Read or write operation to TH4 and TL4 should follow these steps:

Write operation: Low bits first, High bits followed.

Read operation: High bits first, Low bits followed.

Timer4 Modes

Timer4 has two operating modes: 16-bit auto-reload timer and 16 bit auto-reload timer with T4 edge trig. These modes are selected by T4M[1:0] bits in T4CON Register.

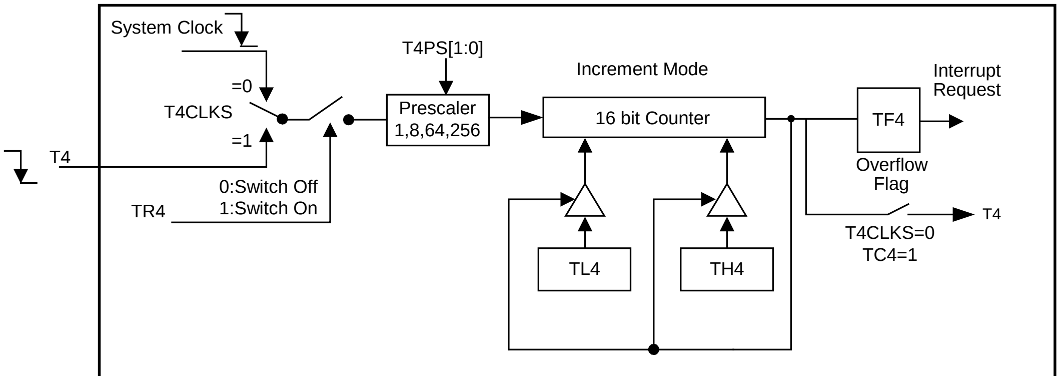
Mode0: 16 bit Auto-Reload Timer

Timer4 operates as 16-bit auto-reload timer in Mode 0. The TH4 register holds the high eight bits of the 16-bit counter/timer, TL4 holds the low eight bits. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the timer overflow flag TF4 (T4CON.7) is set and the 16-bit value in timer load register are reloaded into timer counter register, and an interrupt will occur if Timer 4 interrupts is enabled.

The T4CLKS bit (T4CON.0) selects the counter/timer's clock source. If T4CLKS = 1, external clock from the Pin T4 is selected as Timer4 clock, after prescaled, it will increase the Counter/Timer4 Data register. Else if T4CLKS = 0, the system clock is selected as Timer4 clock.

Setting the TR4 bit (T4CON.1) enables the timer. Setting TR4 does not clear the counter data of Timer4. The timer load register should be loaded with the desired initial value before the timer is enabled.

In Compare mode, the T4 pin is automatically set as output mode by hardware. the internal counter is constantly countered from TH4 and TL4 register value to 0xFFFF. When an overflow occurs, the T4 pin will be inverted. At the same time, interrupt flag bit of Time4 is set. Timer4 must be running in Timer mode (T4CLKS = 0) when compare function enabled.



The Block Diagram of Mode 0 of Timer 4

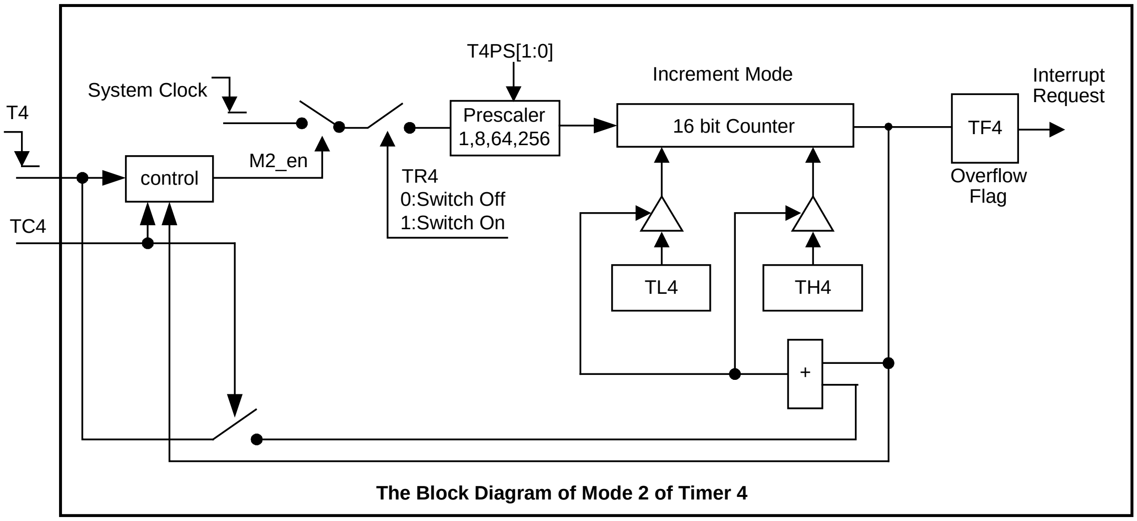
Mode1: 16 bit Auto-Reload Timer with T4 Edge Trig

Timer4 operates as 16-bit timer in Mode1. T4CLKS bit in T4CON.0 will be 0 always.Timer4 can select system clock as clock source. Other setting accords with mode 0.

In Mode1, After Setting the TR4 bit (T4CON.1), Timer4 does not start counting but waits the trig signal (rising or falling edge controlled by T4M[1:0]) from T4. An active trig signal will start the Timer4. When Timer 4 overflows from 0XFFFF to 0x0000, TF4 will be set, TH4 and TL4 will be reloaded from timer load register, and Timer4 holds and waits the next trig edge.

When Timer4 is working, an active trig signal maybe come, if TC4 = 0, the trig signal will be ignored; if TC4 = 1, Timer4 will be re-trigged.

Setting TR4 does not clear the counter data of Timer4. The timer register should be loaded with the desired initial value before the timer is enabled.



The Block Diagram of Mode 2 of Timer 4

control : M2_en set to 1 when T4 edge trig, M2_en set to 0 when counter overflow

Note: *When Timer4 is used as a counter, the frequency of input signal of T4 pin must be less than half of system clock.*

Registers

Table 7.35 Timer4 Control Register

C8H, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T4CON	TF4	TC4	T4PS1	T4PS0	T4M1	T4M0	TR4	T4CLKS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**TF4**	**Timer4 overflow flag bit** 0: No overflow (cleared by hardware) 1: Overflow (Set by hardware)
6	**TC4**	**Compare function Enable bit** When T4M[1:0] = 00 0: Disable compare function of Timer4 1: Enable compare function of Timer4 When T4M[1:0] = 10 or 11 0: Timer4 can't be re-trigged 1: Timer4 can be re-trigged
5-4	**T4PS[1:0]**	**Timer4 input clock Prescale Select bits** 00: 1/1 01: 1/8 10: 1/64 11: 1/256
3-2	**T4M[1:0]**	**Timer4 Mode Select bit** 00: Mode0, 16-bit auto-reload timer 01: reserved 10: Mode2 with rising edge trig from pin T4 (system clock only, T4CLKS is invalid) 11: Mode2 with falling edge trig from pin T4 (system clock only, T4CLKS is invalid)
1	**TR4**	**Timer4 start/stop control bit** 0: Stop Timer4 1: Start Timer4
0	**T4CLKS**	**Timer4 Clock Source select bit** 0: System clock, T4 pin is used as I/O port 1: External clock from pin T4 (On the falling edge), the internal pull-up resister is turned on

Table 7.36 Timer4 Reload/Counter Data Registers

CCH-CDH, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL4	TL4.7	TL4.6	TL4.5	TL4.4	TL4.3	TL4.2	TL4.1	TL4.0
TH4	TH4.7	TH4.6	TH4.5	TH4.4	TH4.3	TH4.2	TH4.1	TH4.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**TL4.x**	Timer4 Low & High byte counter, x = 0 - 7
	TH4.x	

7.9 Interrupt

7.9.1 Feature

- 14 interrupt sources
- 4 interrupt priority levels

The SH79F3218 provides total 14 interrupt sources: 3 external interrupts (INT2, INT3, INT4), 5 timer interrupts (Timer0,1,2, 3, 4), EUART interrupt, TWI interrupt, TK interrupt, PWM1/CRC interrupts, PWM2 Interrupt, SCM interrupt / LPD interrupt.

7.9.2 Interrupt Enable Control

Each interrupt source can be individually enabled or disabled by setting or clearing the corresponding bit in the interrupt enable registers IEN0 or IEN1. The IEN0 register also contains global interrupt enable bit, EA, which can enable/disable all the interrupts at once. Generally, after reset, all interrupt enable bits are set to 0, which means that all the interrupts are disabled.

7.9.3 Register

Table 7.37 Primary Interrupt Enable Register

A8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EA	-	ET2	ES	ET1	EPWM2	ET0	TKIE
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	-	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**EA**	**All interrupt enable bit** 0: Disable all interrupt 1: Enable all interrupt
5	**ET2**	**Timer2 overflow interrupt enable bit** 0: Disable Timer2 overflow interrupt 1: Enable Timer2 overflow interrupt
4	**ES**	**EUART interrupt enable bit** 0: Disable EUART interrupt 1: Enable EUART interrupt
3	**ET1**	**Timer1 overflow interrupt enable bit** 0: Disable Timer1 overflow interrupt 1: Enable Timer1 overflow interrupt
2	**EPWM2**	**PWM2 interrupt enable bit** 0: Disable PWM2 interrupt 1: Enable PWM2 interrupt
1	**ET0**	**Timer0 overflow interrupt enable bit** 0: Disable Timer0 overflow interrupt 1: Enable Timer0 overflow interrupt
0	**TKIE**	**Touch Key interrupt enable bit** 0: Disable Touch Key interrupt 1: Enable Touch Key interrupt

Table 7.38 Secondary Interrupt Enable Register

A9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN1	ESCM/ ELPD	EX3	EPWM1/ ECRC	ET3	ETWI	EX4	EX2	ET4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**ESCM/ELPD**	**SCM/LPD interrupt enable bit** 0: Disable SCM/LPD interrupt 1: Enable SCM/LPD interrupt
6	**EX3**	**External interrupt3 enable bit** 0: Disable external interrupt3 1: Enable external interrupt3
5	**EPWM1/ECRC**	**PWM1/CRC interrupt enable bit** 0: Disable PWM1/CRC interrupt 1: Enable PWM1/CRC interrupt
4	**ET3**	**Timer3 overflow interrupt enable bit** 0: Disable timer3 overflow interrupt 1: Enable timer3 overflow interrupt
3	**ETWI**	**TWI interrupt enable bit** 0: Disable TWI interrupt 1: Enable TWI interrupt
2	**EX4**	**External interrupt4 enable bit** 0: Disable external interrupt4 1: Enable external interrupt4
1	**EX2**	**Enternal interrupt2 enable bit** 0: Disenable external interrupt2 1: Enable external interrupt2
0	**ET4**	**Timer4 overflow interrupt enable bit** 0: Disable Timer4 overflow interrupt 1: Enable Timer4 overflow interrupt

Note:

(1) To enable External interrupt0/1/2/3/4, the corresponding port must be set to input mode before using it.

(2) EPWM is the both PWM1 and PWM2 interrupt enable bit. To enable PWM timer interrupt, the EPWM bit here should be set. Also, the PWMxIE (x = 1, 2) in PWM interrupt control register should be set.

Table 7.39 Interrupt channel Enable Register

BAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IENC	EXS47	EXS46	EXS45	-	-	-	-	EXS40
R/W	R/W	R/W	R/W	-	-	-	-	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	-	-	-	-	0

Bit Number	Bit Mnemonic	Description
7-5, 0	**EXS4x** **（x = 0, 5-7）**	**xternal interrupt 4 channel select bit （x = 0，5-7）** 0: Disable external interrupt 4x 1: Enable external interrupt 4x

Note: *The register is not defined by the blank space must be clear 0.*

Table 7.40 Interrupt channel Enable Register1

BBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IENC1	-	-	ECRC	EPWM1	EPWM2	-	ESCM1	ELPD
R/W			R/W	R/W	R/W	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)			0	0	0	-	0	0

Bit Number	Bit Mnemonic	Description
5	**ECRC**	**CRC interrupt enable bit** 0: Disable CRC interrupt 1: Enable CRC interrupt
4	**EPWM1**	**PWM1 interrupt enable bit** 0: Disable PWM1 interrupt 1: Enable PWM1 interrupt
3	**EPWM2**	**PWM2 interrupt enable bit** 0: Disable PWM2 interrupt 1: Enable PWM2 interrupt
1	**ESCM1**	**SCM interrupt enable bit** 0: Disable SCM interrupt 1: Enable SCM interrupt
0	**ELPD**	**LPD interrupt enable bit** 0: Disable LPD interrupt 1: Enable LPD interrupt

Table 7.41 nterrupt channel Enable Register2

BCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IENC2	-	-	-	-	-		EXS31	EXS30
R/W	-	-	-	-	-		R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-		0	0

Bit Number	Bit Mnemonic	Description
2-0	**EXS3x** **（x = 0-1）**	**xternal interrupt 3 channel select bit (x = 0-1)** 0: Disable external interrupt 3x 1: Enable external interrupt 3x

Note: *The register is not defined by the blank space must be clear 0.*

7.9.4 Interrupt Flag

Each Interrupt source has its own interrupt flag, when interrupt occurs, corresponding flag will be set by hardware, the interrupt flag bits are listed in interrupt abstract table.

When an external interrupt **INT2/3/4** is generated, if the interrupt was edge trigged, the flag IEx (x = 2-4) that generated this interrupt is cleared by hardware when the service routine is vectored. If the interrupt was level trigged, then the requesting external source directly controls the request flag, rather than the on-chip hardware.

The **Timer0/1 interrupt** is generated when they overflows, the flag (TFx, x = 0, 1) in TCON register, which is set by hardware, and will be automatically be cleared by hardware when the service routine is vectored.

The **Timer2** interrupt is generated by the logical OR of flag TF2 in T2CON register, which is set by hardware. None of these flags can be cleared by hardware after CPU responses to the interrupt, the flag must be cleared by software

When the **Timer3** counter overflow, set interrupt flag bit TF3 in T3CON to 1 to generate Timer3 interrupt. The flag will be cleared automatically by hardware after CPU responses to the interrupt.

When the **Timer4** counter overflow, set interrupt flag bit TF4 in T4CON to 1 to generate Timer4 interrupt. The flag will be cleared automatically by hardware after CPU responses to the interrupt.

The **EUART interrupt**s is generated by the logical OR of flag RI and TI in SCON register, which is set by hardware. Neither of these flags can be cleared by hardware when the service routine is vectored. In fact, the service routine will normally have to determine whether it was the receive interrupt flag or the transmission interrupt flag that generated the interrupt, so the flag must be cleared by software.

The **SCM interrupt** is generated by SCMIF in CLKCON register, which is set by hardware. And the flag can only be cleared by hardware.

The **LPD interrupt** is generated by LPDF in LPDCON register. And the flag can only be cleared by hardware. By setting the LPDMD, can choose when the V_{DD} voltage is above or below the LPD set generated when the detecting voltage interruption of LPD.

The **PWM interrupts** are generated by PWMxIF in PWMxC (x = 0, 1). The flags can be cleared by software.

The **CRC interrupt** is generated by CRCIF in CRCCON. The flags can be cleared by software.

The **TK interrupt** is generated by IFERR, IFGO, IFAVE, IFCOUNT, IFTKOV in TKF0. The flags can be cleared by software.

The **TWI interrupt** is generated by TFREE, TWINT, TOUT in TWICON. The flags can be cleared by software.

Table 7.42 External Interrupt Flag Register

E8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXF0	IT4.1	IT4.0	IT3.1	IT3.0	IT2.1	IT2.0	IE3	IE2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-6	**IT4[1:0]**	**External interrupt4 trigger mode selection bit** 00: Low Level trigger 01: Trigger on falling edge 10: Trigger on rising edge 11: Trigger on both edge IT4 [1:0] is effect on external interrupt 4x at the same mode
5-4	**IT3[1:0]**	**External interrupt3 trigger mode selection bit** 00: Low Level trigger 01: Trigger on falling edge 10: Trigger on rising edge 11: Trigger on both edge
3-2	**IT2[1:0]**	**External interrupt2 trigger mode selection bit** 00: Low Level trigger 01: Trigger on falling edge 10: Trigger on rising edge 11: Trigger on both edge
1	**IE3**	**External interrupt3 request flag bit** 0: No interrupt pending 1: Interrupt is pending
0	**IE2**	**External interrupt2 request flag bit** 0: No interrupt pending 1: Interrupt is pending

Table 7.43 External Interrupt 4 Flag Register

D8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXF1	IF47	IF46	IF45	-	-	-	-	IF40
R/W	R/W	R/W	R/W					R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0					0

Bit Number	Bit Mnemonic	Description
7-5, 0	**IF4x** **（x = 0, 5-7）**	**External interrupt4 request flag bit** 0: No interrupt pending 1: Interrupt is pending IF4x is cleared by software

Note: *The register is not defined by the blank space must be clear 0.*

Table 7.44 External Interrupt 3 Flag Register

D1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXF3	-	-	-	-	-	-	IF31	IF30
R/W	-	-	-	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-	-	0	0

Bit Number	Bit Mnemonic	Description
2-0	**IF3x （x = 0-1）**	**External interrupt3 request flag bit** 0: No interrupt pending 1: Interrupt is pending IF3x is cleared by software

Note: *The register is not defined by the blank space must be clear 0.*

7.9.5 Interrupt Vector

When an interrupt occurs, the program counter is pushed onto the stack and the corresponding interrupt vector address is loaded into the program counter. The interrupt vector addresses are listed in **Interrupt Summary table**.

7.9.6 Interrupt Priority

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing corresponding bits in the interrupt priority control registers IPL0, IPH0, IPL1, and IPH1. The interrupt priority service is described below.

An interrupt service routine in progress can be interrupted by a higher priority interrupt, but can not by another interrupt with the same or lower priority.

The highest priority interrupt service cannot be interrupted by any other interrupt source. If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced.

If the same priority interrupt source apply for the interrupt at the beginning of the instruction cycle at the same time, an internal polling sequence determines which request is serviced.

Interrupt Priority		
Priority bits		**Interrupt Lever Priority**
IPHx	IPLx	
0	0	Level 0 (lowest priority)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest priority)

Table 7.45 Interrupt Priority Control Registers

B8H, B4H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
IPL0	-	-	PT2L	PESL	PT1L	PW2L	PT0L	PTKL
IPH0	-	-	PT2H	PESH	PT1H	PW2H	PT0H	PTKH
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	0	0	0	0	0	0
B9H, B5H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
IPL1	PSCL	PX3L	PW1L	PT3L	PTWL	PX4L	PX2L	PT4L
IPH1	PSCH	PX3H	PW1H	PT3H	PTWH	PX4H	PX2H	PT4H
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	**Bit Mnemonic**	**Description**
7-0	**PxxxL/H**	Corresponding interrupt source xxx's priority level selection bits

Note: *The register is not defined by the blank space must be clear 0.*

7.9.7 Interrupt Handling

The interrupt flags are sampled and captured at each machine cycle. All interrupts are sampled at the rising edge of the clock. If one of the flags was set, the CPU will find it and the interrupt system will generate a LCALL to the appropriate service routine, LCALL generated by hardware is not blocked by any of the following conditions:

An interrupt of equal or higher priority is already in progress.

The current cycle is not in the final cycle of the instruction in progress. In other words, any interrupt request can not get response before executing instructions to complete.

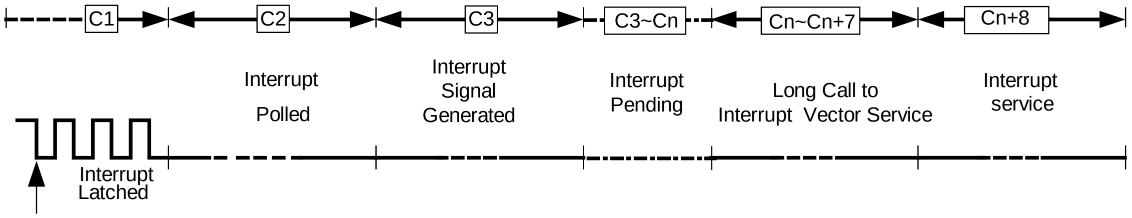
The instruction in progress is RETI or visit the special register IEN01 or IPLH instruction. This ensures that if the instruction in progress is RETI or read and write IEN01 or IPLH then at least one more instruction except RETI will be executed before any interrupt is vectored to; this delay guarantees that the CPU can observe the changes of the interrupt status.

Note:

Since priority change normally needs 2 instructions, it is recommended to disable corresponding Interrupt Enable flag to avoid interrupt between these 2 instructions during the change of priority.

If the flag is no longer active when the blocking condition is removed, the denied interrupt will not be serviced. Every polling cycle interrogates only the valid interrupt requests.

The polling cycle/LCALL sequence is illustrated below:



Interrupt Response Time

The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW). Then vector address of the corresponding interrupt source (referring to the interrupt vector table) will be stored in the program counter.

Interrupt service execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that the interrupt routine is no longer in progress, and then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from the point where it was stopped. The RET instruction can also return to the original address to continue, but the interrupt priority control system still think the interrupt in a same priority is responsed, in this case, the same priority or lower priority interrupt will not be responsed.

7.9.8 Interrupt Response Time

If an interrupt is detected, its request flag will be set in every machine cycle after detection. The value will be kept by the internal circuitry until the next machine cycle; the CPU will generate an interrupt at the third machine cycle. If the request is active and conditions are right for it to be acknowledged, hardware instruction LCALL will call service routine which requeste interrupt at the next instruction to be executed. Otherwise the interrupt will pending. The call itself takes 7 machine cycles. Therefore, from the external interrupt request to start the implementation of interrupt program requires at least 3+7 completed machine cycle.

A longer response time would be obtained if the request was blocked by one of the above three previously listed conditions. If an interrupt of equal or higher priority is already in progress, the additional wait time obviously depends on the length of the other interrupt’s service routine.

If the instruction in progress is not in its final cycle and the instruction in progress is RETI, the additional wait time is 8 machine cycles. For a single interrupt system, if the next instruction is 20 machine cycles long (the longest instructions DIV & MUL are 20 machine cycles long for 16 bit operation), adding the LCALL instruction 7 machine cycles the total response time is 2+8+20+7 machine cycles.

Thus interrupt response time is always more than 10 machine cycles and less than 37 machine cycles.

7.9.9 External Interrupt Inputs

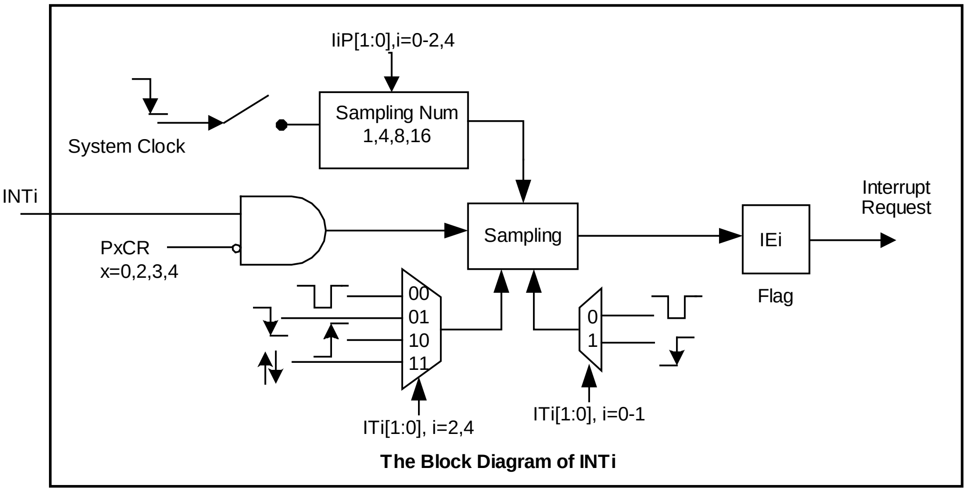
The SH79F3218 has 3 external interrupt inputs. External interrupt 2 each has one vector address. External interrupt 4 has 4 inputs, all of them share one vector address. External interrupt 3 has 3 inputs, all of them share one vector address. External interrupt 2/3/4 can be set up by setting the EXF0 register ITx (x = 2, 3, 4), which is the level of the trigger or the edge of the trigger or a double edge trigger. When ITx = 00 (x = 2, 3, 4), the external interrupt INTx (x = 3, 2, 4) pin is low level trigger; when ITx (x = 3, 2, 4) = 01, 10, external interrupt INTx (x = 4, 3) is a falling edge. As the external interrupt pin is sampled at each cycle, the input high or low level should be maintained at least SN cycles to ensure that the correct sampling can be sampled. When ITx (x = 2, 3, 4) = 11, the external interrupt INTx (x = 3, 4) is a double edge trigger, and any level of conversion will trigger an interrupt request.

If the external interrupt is a falling edge or the rising edge is triggered, an external interrupt source should be kept at least a SN cycle high (low) level, and then at least a SN cycle low (high) level. This ensures that the edge can be detected in order to make the IEx 1. After the call interrupt service routine, CPU automatically will IEx 0.

If an external interrupt is triggered by a low level, the external interrupt source must remain valid until the request is interrupted, and the process requires 2 times the SN sampling period. If the interrupt service is completed and the external interrupt is maintained, the next interrupt will be generated. Interrupt flag IEx (x = 2, 3, 4) when the interrupt level is triggered, because the interrupt is only related to the input level.

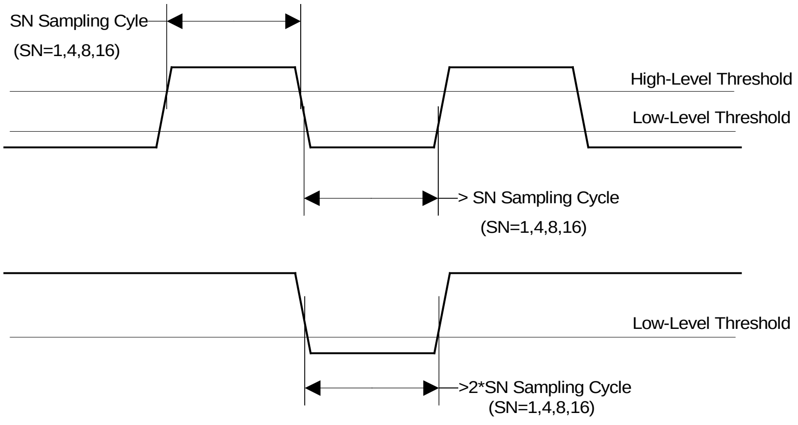
Sample clock Prescaler Select bits and sample times Select bits in **EXCON** register.

When SH79F3218 is in IDLE mode or Power-Down mode, interrupt will cause the processor to wake up and resume operation, refer to “**Power Management**” chapter for details.



The Block Diagram of INTi

***Note**: IE2 is automatically cleared by CPU when the service routine is called while IF40,45-47and IF30-32 should be cleared by software.*



External Interrupt Detection

Table 7.46 External interrupt sample times control registrer

A4H, Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXCON	I1PS1	I1PS0	I1SN1	I1SN0	I0PS1	I0PS0	I0SN1	I0SN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-6	**I1PS[1:0]**	**INT4 sample clock Prescaler Select bits** 00: 1/1 01: 1/16 10: 1/64 11: 1/128
5-4	**I1SN[1:0]**	**INT4 sample times Select bits** 00: 1 01: 2 10: 3 11: 4
3-2	**I0PS[1:0]**	**INT2/3 sample clock Prescaler Select bits** 00: 1/1 01: 1/16 10: 1/64 11: 1/128
1-0	**I0SN[1:0]**	**INT2/3 sample times Select bits** 00: 1 01: 2 10: 3 11: 4

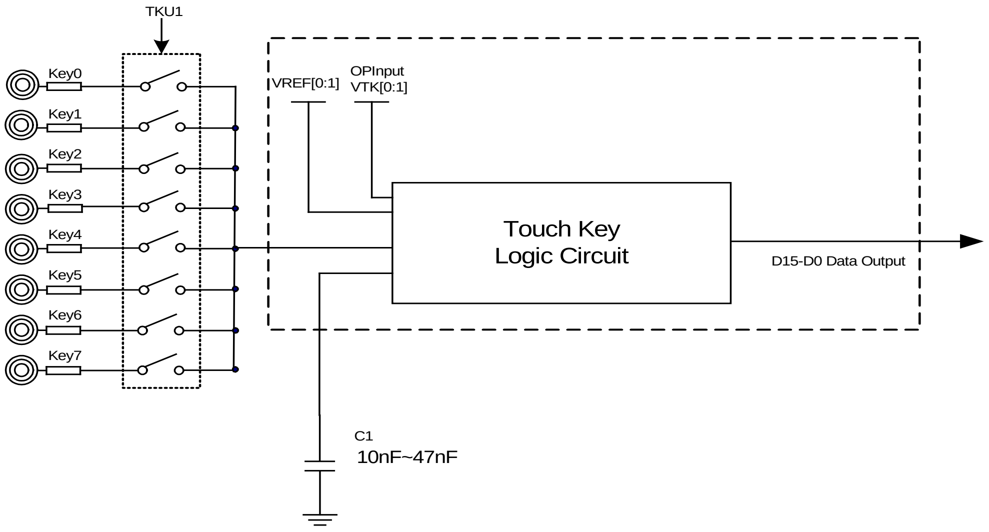
Note: *If I0SN[1:0] = 11, the INT2, 3 (falling edge tiggle) need sample 4 times tah can be set the interrupt flag.*

7.9.10 Interrupt Summary

Source	Vector Address	Enable bits	Flag bits	Polling Priority	Interrupt number (C51)
Reset	0000H	-	-	0 (highest)	-
TK	0003H	TKIE	IFERR+IFGO +IFAVE+IFCOUNT +IFTKOV	1	0
Timer0	000BH	ET0	TF0	2	1
PWM2	0013H	EPWM2	PWM02IF	3	2
Timer1	001BH	ET1	TF1	4	3
EUART	0023H	ES	RI+TI	5	4
Timer2	002BH	ET2	TF2+EXF2	6	5
Timer4	003BH	ET4	TF4	8	7
INT2	0043H	EX2	IE2	9	8
INT4	004BH	EX4+EXF1	IF47-40	10	9
TWI	0053H	ETWI	TWINT	11	10
Timer3	005BH	ET3	TF3	12	11
PWM1/CRC	0063H	EPWM1/ECRC	PWM01IF/CRCIF	13	12
INT3	006BH	EX3	IF32-IF30	14	13
SCM/LPD	0073H	ESCM+ESCM1/ELPD	SCMIF/LPDF	15	14

8. Enhanced Function

8.1 Touch Key Function



System Diagram

Functional Description

SH79F3218 built-in Touch Key function module, which can connect at most 8 keys .

SH79F3218 built-in simplified operating circuit in Touch Key function module, the application of it only need to use a external connected C1 capacitor. The value of C1 capacitor choose 22 nF - 44nF, must use polyester capacitor of 10% or more accuracy, X7R capacitor or NPO capacitor. C1 capacitor can adjust suitable sensitivity according to the material of actual circuit or medium of Touch Key, the value of capacitor is smaller, the sensitivity is higher, the value of capacitor is higher, the sensitivity is smaller,

Touch Key module can select a scan button channel number by TKU1 registers, which can connect at most 8 keys.

If Touch Key function is not used, it can be set as I/O ports, SEG output or COM output through register.Refer to “I/O Port” chapter for details.

On-off circuit can be selected by FSW1 bit and FSW0 bit in register. Working frequency recommend select 4M or below 4M. Touch Key built-in reference voltage, can be selected by VREF1 bit and VREF0 bit in register.

Touch Key make sure the stability of data register in dirrerent C1 and working frequency by adjusting TUNE1 bit and TUNE0 bit.

According to the actual application, Touch Key sampling times can select multi-sampling. Program only need to start one time sampling scan, hardware will perform multi-sampling, count the average value and output result automatically. For example, selecting 6 times sampling output, set TKGO/$\overline{\text{DONE}}$ bit and start to scan key, hardware will sample 6 times value and campare the six samplied value.Removing maximum and minimum, the rest of them will be averaged to output to16-bit data register.

28-bit amplification coefficient register is used to amplify the calculated result of key controller. If calculated result is larger than 16-bit data, that is to say calculated result is high-bit overflow.The flag bit IFERR will be set. If interrup enables, interrupt program will be called. At the time, user need to reduce data value of amplification coefficient register and restart next scan. Usually during normal operation, the value of 16-bit data register will not be larger than FFFFH. When the value is larger than FFFFH, data value of dividend register will be reduced.

There are five kinds of touch buttons will produce the interrupt flag, in which the 1th-4th will be interrupted, 5th is only interrupt flag, but not in response to interrupt, need to judge the interrupt flag bit after the implementation of interrupt subroutine:

(1) after the end of the scan button, if the result of the operation is high, the interrupt flag bit IFERR is set to 1. If it is a multi - sampling, the system will stop the current sampling status and wait for the next restart, and do not perform the sampling. If the result of the operation is high, the user should reduce the 28 bit value of the amplifier.

(2) if TKGO/$\overline{\text{DONE}}$=1, the system automatically detects the output status is normal, if the exception will flag bit IFGO 1, at this time to start the error, the user should delay 10uS, restart the next scan.

(3) after the scan is finished, if an exception is not an exception, the interrupt flag bit IFAVE is set to 1.

(4) in the process of counting the scan button, when the count register overflows, the interrupt flag bit IFCOUNT is set to 1, the user needs to reduce the capacitor or the 28 bit of the value of the amplification factor.

(5) in the LED, set the TKCOUNT (touch button time width), start the touch button, if the TKCOUNT time, the touch button is not yet over, the sign bit IFTKOV set 1, but scan button and LED will continue to scan and will not be interrupted, users need to determine the IFTKOV flag in case of interruption, adjust the TKCOUNT time to ensure that the touch key scan time.When used as Touch Key, the step of startup as follows:

(1) Select key channel which is needed to scan;

(2) Set TKCON bit, enable Touch Key module works;

(3) Set switch frequency, reference voltage (Vref), key sampling times and the sequence of scan;

(4) Set 28-bit amplification coefficient register;

(5) Software delay 10uS;

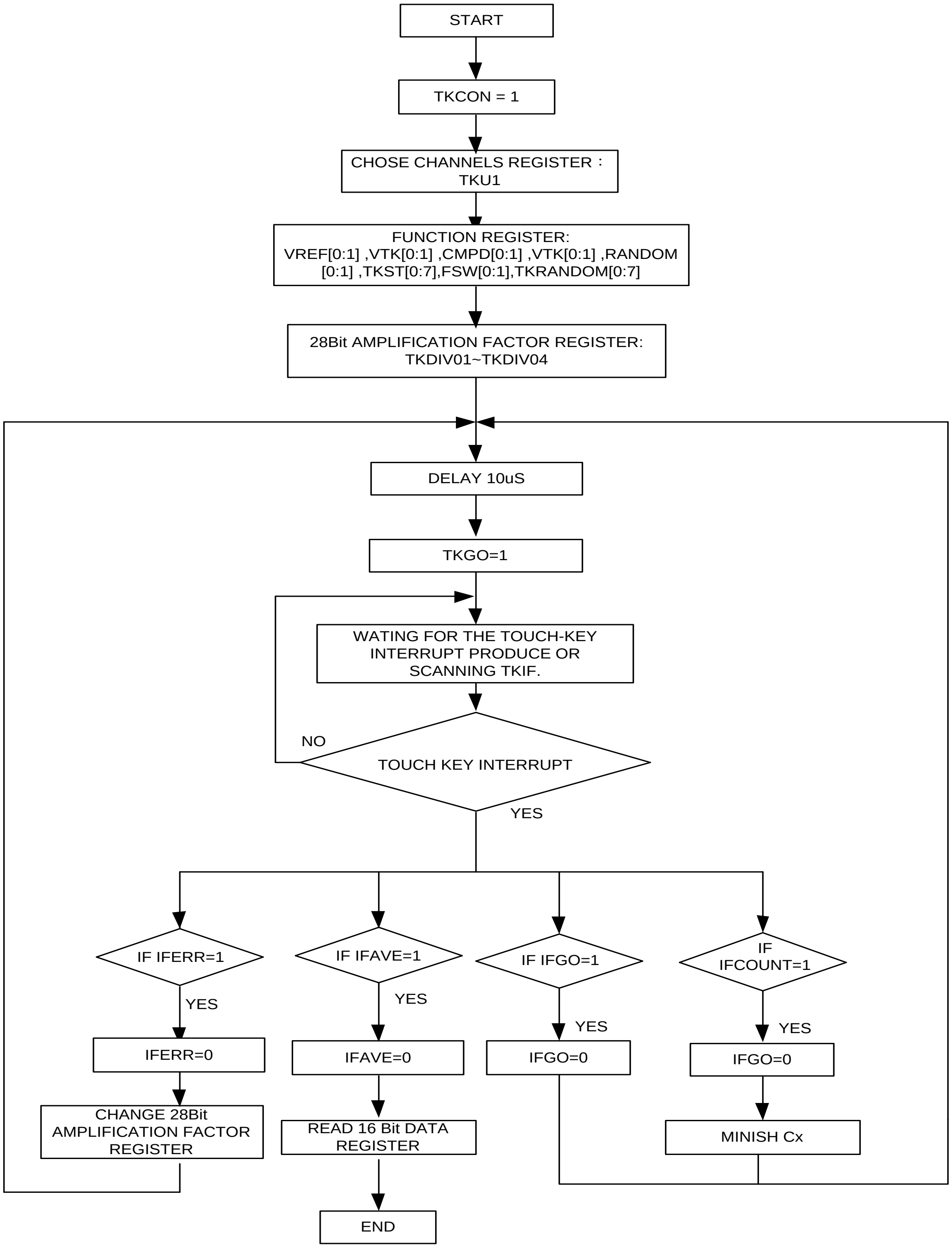
(6) Set TKGO/$\overline{\text{DONE}}$ bit, start key scan;

(7) Interrupt generates, TKGO will be cleared by hardware automatically;

(8) Judging interrupt flag bit: IFERR, IFGO, IFAVE, IFCOUNT;
If IFAVE = 1, reading data register 500H – 50FH, program save data result, goto step 9 ;
If IFERR = 1, data register arithmetic overflow error, clear IFERR flag bit, reset amplification coefficient register, reduce the value of amplification coefficient, goto step5 to restart scan;
If IFGO = 1, key controller startup errors, clear IFGO flag bit, goto step 5 to restart scan (channel flag register is invalid);
If IFCOUNT = 1, key scan count overflow error, clear IFCOUNT flag bit, reduce C1 capacitor or slow down switch frequency. Goto step 5 to restart scan.

(9) Completing A group key scan.

Operating Flow



8.1.1 Register

Table 8.1 Touch Key Functional Control Register

A1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCON1	TKCON	-	TKGO /$\overline{\text{DONE}}$	SHARE	MODE	OVDD	FSW1	FSW0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	-	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**TKCON**	**Touch Key Enable bit** 0: Disable Touch Key Function 1: Enable Touch Key Function
5	**TKGO/$\overline{\text{DONE}}$**	**Start Touch Key Enable bit** 0: Don't start key scan or key scan end 1:Start key scan or key scan is performing
4	**SHARE**	**Touch Key share with LED Enable bit (refer to Note3 for details)** 0: Disable sharing 1: Enable sharing
3	**MODE**	**Touch Key Mode Select bit** 0: Select battery charging times as data parameter 1: Select battery charging time as data parameter
2	**OVDD**	**OP Output V_{DD} Enable bit** 0: Select OP output voltage 1: Select V_{DD} output voltage
1-0	**FSW[1:0]**	**Sampling Times Selection bit** 00: Key sample 1 time output data, D15-D0 defined as the average of sampling 1 time 01: Key sample 3 times output data, D15-D0 defined as the average of sampling 1 time 10: Key sample 6 times output data, D15-D0 defined as the average of sampling 4 times 11: Key sample 10 times output data, D15-D0 defined as the average of sampling 8 times (Except for maximum and minimum)

Note: *If OVDD = 0, OP output voltage is selected by VTK; If OVDD = 1, V_{DD} output the OP output voltage directly.*

Table 8.2 Touch Key Functional Frequency Control Register

CFH，Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKST	-	ST.6	ST.5	ST.4	ST.3	ST.2	ST.1	ST.0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
6-0	**ST[6:0]**	**Touch Key Functional Frequency Control bit** System clock/TKST = Touch Key Functional On-Off Frequency

Note: *Touch Key Functional Frequency = OSC/TKST; The range of TKST is 2 - 127 frequency diision, when register is less than or equal to 2 frequency diision, register is system ckock/2 by default.*

Table 8.3 Touch Key Frequency Random Setting Register

B6H，Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKRANDOM	TKRADON	TKOFFSET	TKVDD	TKOUT	-	-	RANDOM1	RANDOM0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	-	-	0	0

Bit Number	Bit Mnemonic	Description
7	**TKRADON**	**Touch Key Random Frequency Enable bit** 0: Disable Touch Key random frequency function 1: Enable Touch Key random frequency function
6	**TKOFFSET**	**Touch Key Offset Compensation Enable bit** 0: Disable Touch Key offset compensation 1: Enable Touch Key offset compensation
5	**TKV$_{DD}$**	**Touch Key Compensation Waveform Level Selection bit** 0: OP output Touch Key Compensation Waveform Level 1: V$_{DD}$ output Touch Key Compensation Waveform Level
4	**TKOUT**	**Touch Key Compensation Waveform Output ability Selection bit** 0: Touch Key Compensation Waveform weak output 1: Touch Key Compensation Waveform strong output
1-0	**RANDOM[1:0]**	**Random Shake Setting bit** **TKRADSEL = 0** 00: TKST random shake ±1 01: TKST random shake ±1, ±2 10: TKST random shake ±1, ±2, ±3 11: TKST random shake ±1, ±2, ±3, ±4

Note: *Design spec: random shake please make sure that mathematic accumulation is 0 during a certain period of time.*

(1) When TKST is two divided-frequency,can't shake clock, three divided-frequency is valid only when selecting 00, four divided-frequency valid option is 00, 01. When TKST is more than or equal to six divided-frequency, TKST divided-frequency valid option is 00, 01, 10, 11.

(2) If Touch Key offset compensation bit is valid, when Touch Key scans, other TK scanning key channel all output compensation waveform except for the current TK scanning key channel.
When TKVDD is valid, V$_{DD}$ provides Touch Key Compensation Waveform Level.
When TKVDD is 0, OP output provides Touch Key Compensation Waveform Level.

Table 8.4 Touch Key Interrupt Flag Register (The register only can be cleared)

C1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKF0	-	IFERR	IFGO	IFAVE	IFCOUNT	IFTKOV	-	-
R/W		R/W	R/W	R/W	R/W	R/W	-	-
Reset Value (POR/WDT/LVR/PIN)		0	0	0	0	0	-	-

Bit Number	Bit Mnemonic	Description
6	**IFERR**	**Calculated Result Overflow Interrupt Flag bit** 0: Calculated result high-bit don’t overflow 1: Calculated result high-bit overflow and generate interrupt
5	**IFGO**	**Start signal Error Interrupt Flag bit** 0: Start signal has no error 1: Start signal error generates interrupt
4	**IFAVE**	**Key Scan End Interrupt Flag bit** 0: Scan don’t end 1: Scan end and generate interrupt
3	**IFCOUNT**	**Key Scan Count Overflow Flag bit** 0: Key scan count don’t overflow 1: Key scan count overflow
2	**IFTKOV**	**Error Signal Interrupt Flag bit (SHARE status, LED scan start, TK havn’t completed)** 0: TK scan end normally, LED start normally 1: TK scan time add, LED scan time delay, generate interrupt

Table 8.5 Amplification Coefficient Register

C3H-C6H，Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKDIV01 (C3H)	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
TKDIV02 (C4H)	DIV15	DIV14	DIV13	DIV12	DIV11	DIV10	DIV9	DIV8
TKDIV03 (C5H)	DIV23	DIV22	DIV21	DIV20	DIV19	DIV18	DIV17	DIV16
TKDIV04 (C6H)	-	-	-	-	DIV27	DIV26	DIV25	DIV24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**TKDIV0x** **x = 1- 4,**	**Touch key Amplification Coefficient** DIV0 - DIV27: Fill in a 27 bit value as a touch key, the greater the value, the greater the value obtained.

Table 8.6 Port Function Control Register

9DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2SS	P2SS.7	P2SS.6	P2SS.5	P2SS.4	P2SS.3	P2SS.2	P2SS.1	P2SS.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**PxSSy** **x = 0-5, y = 0-7**	**Port Function Control** 0: PxSSy is I/O 1: PxSSy Is Touch Key channel

Note: *P2SS.1 - P2SS.7 corresponds to TCH1 - TCH7 (P2.0 - P2.6). P2SS.0 corresponds to TCH0 (P1.7).*

Table 8.7 Touch Key Function Time Control Register

E6H, Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKCOUNT	COUNT0.7	COUNT0.6	COUNT0.5	COUNT0.4	COUNT0.3	COUNT0.2	COUNT0.1	COUNT0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
0 **7-0**	**COUNT0[7:0]**	**TK Time Width Selection bit** TK clock width = LED clock frequency * TKCOUNT

Table 8.8 Touch Key Wait Time Control Register

A2H, Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKWAIT	TKWAIT.7	TKWAIT.6	TKWAIT.5	TKWAIT.4	TKWAIT.3	TKWAIT.2	TKWAIT.1	TKWAIT.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
0 **7-0**	**TKWAIT[7:0]**	**TK Wait Time Selection bit** TK wait clock width = LED clock frequency * TKCOUNT

Note: *TKCOUNT ,TKWAIT register valid only in SHARE mode.*

In share mode , LED frame time is divided into three parts, T_A for the duration of the touch key, T_B for the duration of the LED scanning, T_C is touch scan button waiting time width. In the formula below, T_{LED} is LED clock, and T_{SYS} is system clock.

*T_{LED} = TSYS*8*DISPCLK*

*T_A = T_{LED} *TKCOUNT [time A]*

*T_B = (T_{LED} *DISCOM) *S [time B]*

S is the number of LED scanning: scanning 4COM, S = 4 ; S = 5, 5COM scanning, and so on.

*T_C = T_{LED} *TKWAIT [time C]*

T_B + T_C + T_A = 64Hz for example:

In SHARE mode, when the system clock is RC = 16.6MHz, and the frame rate of 64Hz, COM for 5COM scanning:

Well, T_{SYS} = 1/16.6M = 60.24ns

Need results for T_C + T_B + T_A = 1/64s = 15.625ms

If you set DISPCLK to 96H (DISPCLK theory, the smaller the better, but also to ensure that TKCOUNT, DISCOM, TKWAIT will not overflow)

*T_{LED} = T_{SYS} *8*DISPCLK*

*= 60.24ns*8*150*

= 72288ns = 72.288us = 0.072288ms

Need to get the T_A = 7ms touch button time width: (the time width of the touch button is required to ensure that during this time, all the touch keys can be collected to complete)

TKCOUNT = T_A / T_{LED} = 7ms/0.072288ms = 96 = 60H

Need to get T_B = 6.625ms LED scan time width: (LED scan time width needs to be as much as possible, to ensure LED brightness)

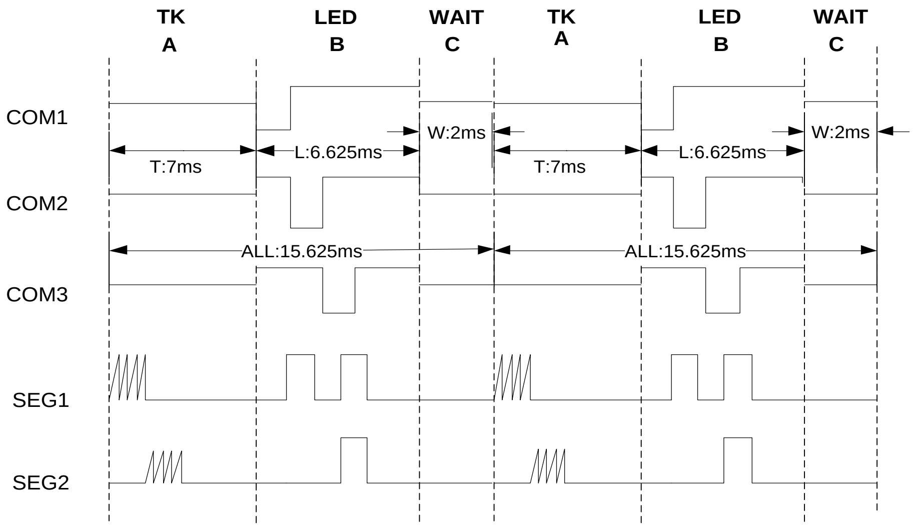
DISCOM = T_B /S/ T_{LED} = 6.625ms/5/0.072288ms = 18 = 12H

Need to get the T_C = 2ms touch button to scan the waiting time width: (touch button scanning wait time width in the case of no impact on the touch keys, the need to minimize the time, generally can be set to 0)

TKWAIT = T_C / T_{LED} = 2ms/0.072288ms = 27 = 1BH

When A, B, C, when there is a period of time if the change ones, other time was no change, frame rate will change.

In the actual use of the process, the touch buttons will change with the temperature and humidity of the environment, so that the time width of the touch button T_A need to set aside a part of the spare time to serve as the margin, usually for all the touch button to complete the time plus 10-15%.



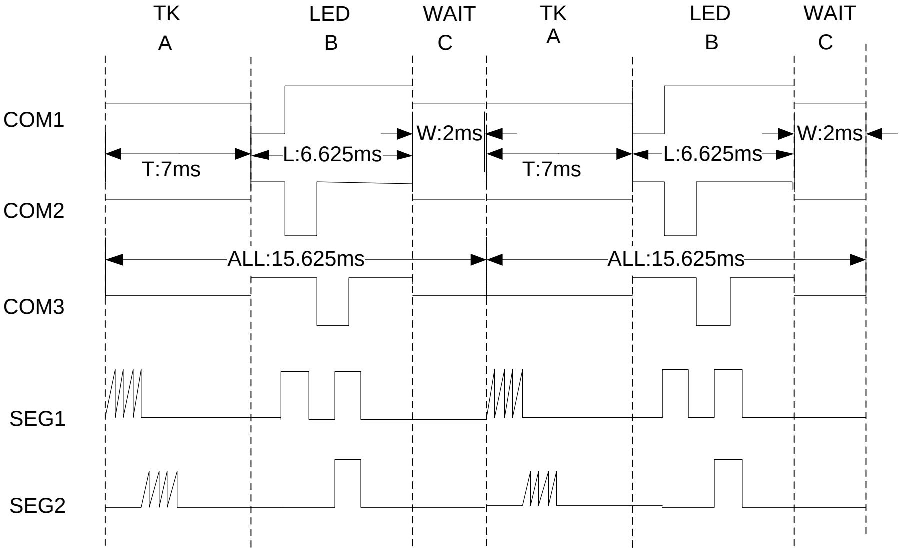
TK & LED (SHARE)

Table 8.9 Key Scan Error Register

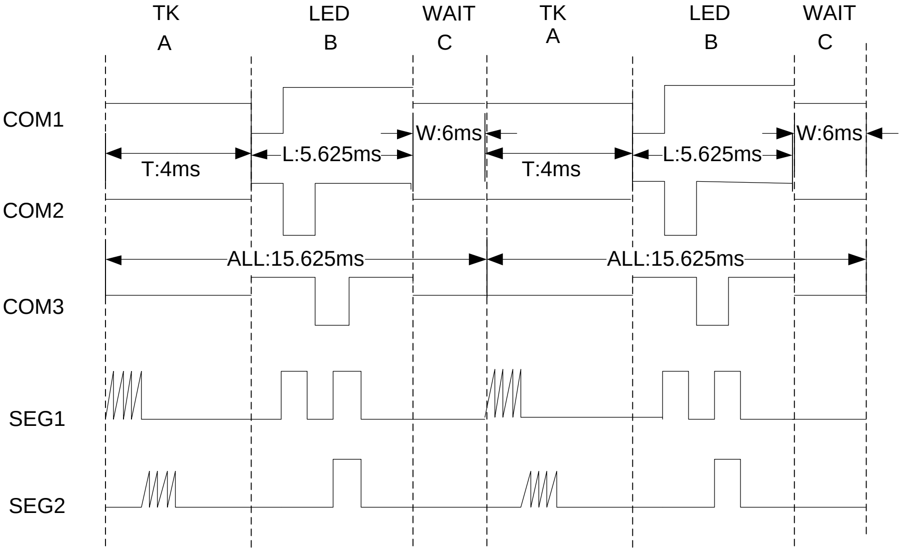
B7H，Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKW	-	-	-	-	-	TW.2	TW.1	TW.0
R/W	-	-	-	-	-	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-	0	0	0

Bit Number	Bit Mnemonic	Description
2-0	**TW[2:0]**	**Key Scan Error Display:** When TKW[2-0] = 000b, Key 1 When TKW[2-0] = 001b, Key 2 When TKW[2-0] = 010b, Key 3 When TKW[2-0] = 011b, Key 4 When TKW[2-0] = 100b, Key 5 When TKW[2-0] = 101b, Key 6 When TKW[2-0] = 110b, Key 7 When TKW[2-0] = 111b, Key 8

Note: *When key error flag bit is set, all other flag bit will stop running Touch Key, except for IFTKOV bit. Error channel bit will be set (When IFGO flag bit is set, key scan error register is invalid). The register is read-only register.*



TK & LED (SHARE) 01



TK & LED (SHARE) 02

SEG output Touch Key waveform in diagram:

ALL width = LED frame frequency width, T width is setting width of Touch Key, L width is width of LED scan, C width is width of LED COM.

ALL (LED frame frequency) = L (LED scan width)+T (Touch Key time width)+W(Touch Key wait time width)

For example:

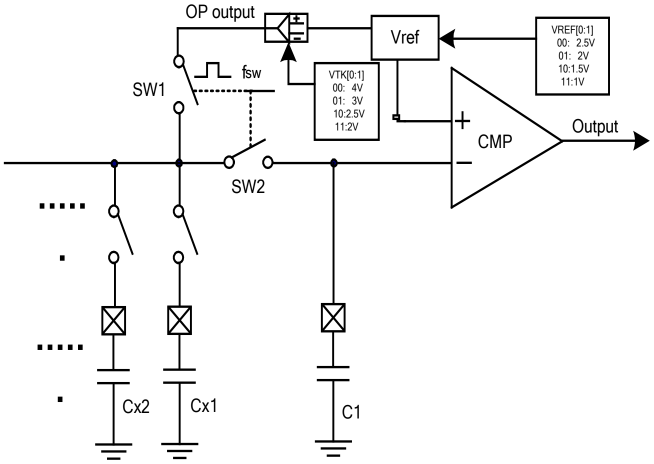
When TK width is 7ms, LED width is 6.625ms,TK wait time width is 2ms,then frame frequency will be 64Hz.

When TK width is 4ms, LED width is 5.625ms, TK wait time width is 6ms,then frame frequency will be 64Hz.

Table 8.10 Reference Voltage Selection Register

CEH，Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKVREF	VREF1	VREF0	CMPD1	CMPD0	VTK1	VTK0	TUNE1	TUNE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-6	**VREF[1:0]**	**Internal Reference Voltage Selection bit** 00: V_{REF} = 2.5V 01: V_{REF} = 2V 10: V_{REF} = 1.5V 11: V_{REF} = 1V
5-4	**CMPD[1:0]**	**Debounce Time Selection bit** 00: About 4 X t_{sysclk} 01: About 8 X t_{sysclk} 10: About 16 X t_{sysclk} 11: About 32 X t_{sysclk}
3-2	**VTK[1:0]**	**OP Output Voltage Selection bit** 00: VTK = 4V 01: VTK = 3.0V 10: VTK = 2.5V 11: VTK = 2V
1-0	**TUNE[1:0]**	**Discharge Time Adjust Selection bit** 00: Delay 128 X t_{sysclk} 01: Delay 256 X t_{sysclk} 10: Delay 384 X t_{sysclk} 11: Delay 512 X t_{sysclk}



OP Output Voltage Diagram

Table 8.11 Key Scan Sequence Register

C2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TKU1	TK8	TK7	TK6	TK5	TK4	TK3	TK2	TK1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**TK[8:1]**	**Key scan sequence control bit** 0: skip scan the key 1: sequential scanning the key

Note: *When some bit of TKU1 is cleared, starting key scan will skip the channel of this key. The sequence of key scan is from TK1 - TK8 , representing 8 channels of key scan separately.*

Table 8.12 16-bit Data Register (Touch Key data RAM is read-only register)

Address		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
500H	**TK01L**	D7	D6	D5	D4	D3	D2	D1	D0
501H	**TK01H**	D15	D14	D13	D12	D11	D10	D9	D8
502H	**TK02L**	D7	D6	D5	D4	D3	D2	D1	D0
503H	**TK02H**	D15	D14	D13	D12	D11	D10	D9	D8
504H	**TK03L**	D7	D6	D5	D4	D3	D2	D1	D0
505H	**TK03H**	D15	D14	D13	D12	D11	D10	D9	D8
506H	**TK04L**	D7	D6	D5	D4	D3	D2	D1	D0
507H	**TK04H**	D15	D14	D13	D12	D11	D10	D9	D8
508H	**TK05L**	D7	D6	D5	D4	D3	D2	D1	D0
509H	**TK05H**	D15	D14	D13	D12	D11	D10	D9	D8
50AH	**TK06L**	D7	D6	D5	D4	D3	D2	D1	D0
50BH	**TK06H**	D15	D14	D13	D12	D11	D10	D9	D8
50CH	**TK07L**	D7	D6	D5	D4	D3	D2	D1	D0
50DH	**TK07H**	D15	D14	D13	D12	D11	D10	D9	D8
50EH	**TK08L**	D7	D6	D5	D4	D3	D2	D1	D0
50FH	**TK08H**	D15	D14	D13	D12	D11	D10	D9	D8

Note:

(1) OP output voltage is supply voltage of Touch Key, Vref is reference voltage source of Touch Key.

(2) Touch Key can adjust discharge time of capacitor by setting TUNE1 bit and TUNE0 bit to ensure the stability of data register under the condition of different Cx and working frequency.

(3) When enable LED share with Touch Key: TK1 - TK8 will be used as key, function scan will exit Touch Key module.It is need to set SEG and COM before enabling LED display module. The content of scanning COM1 - COM7, SEG0-7, SEG11-15 need to be prestored. LED function is finished and restart Touch Key function after a scanning.

8.2 LED Driver

LED driver contains a controller, 7 Common signal pins, and 13 Segment driver pins. Support 1/1 - 1/7 duty voltage driver mode. The drive mode is selected by LEDCOM register. The controller is composed of the display data RAM storage area and a duty cycle generator.

LED_SEG0-7, 11-15 pin can be used as a I/O port, the P0SS and P2SS registers are set to I/O. SEG01, SEG02 registers are used to control the LED_SEG0-7, 11-15, and LEDCOM registers for the control of COM1-COM7 and I/O port mode selection.

LED is shut down during power on reset, pin reset, low voltage reset or watchdog reset. LEDMODE(0,1) decision led output common cathode mode or common anode mode. When LEDMODE(0,1) = 00 ,common to a low level, the output high level segment guide led, when LEDMODE (0,1)= 01, common output high level, segment are output low level conductivity led.

LED is shut down during power on reset, pin reset, low voltage reset or watchdog reset.

8.2.1 Register

Table 8.13 LED Control Register

D9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DISPCON	-	LEDON	-		LEDMODE1	LEDMODE0		MODSW
R/W	-	R/W	-		R/W	R/W		R/W
Reset Value (POR/WDT/LVR/PIN)	-	0	-		0	0		0

Bit Number	Bit Mnemonic	Description
6	**LEDON**	**LED control bit** 0: Disable LED driver 1: Enable LED driver
3-2	**LEDMODE[1:0]**	**LED mode control bit** 00: common anode mode 01: common cathode mode 1X: common cathode/anode mode
0	**MODSW**	**LED shared control bit** 0: SEG0x, LEDCOM is valid 1: all the LED Pins worked as IO

Note:

LED display module provides three kinds of mode selection, when the common cathode mode, *led ram COMxL (x = 1 - 7) effective, other invalid; when the* common anode mode, *the LED ram COMxH (x = 1 - 7) and other invalid; when the model for the common cathode/anode mode, led the ram is effective and COMxL (x = 1 - 7) as a common cathode mode ram, COMxH (x = 1 - 7) as common anode mode of ram.*

In share mode, and the touch buttons at the same time, only for common cathode mode*.*

The MODSW bit controls the shared bits of all LED pins;

If MODSW = 1, then all SEG/COM LED ports are IO;

If MODSW = 0, then all the SEG/COM LED ports do for the show, continue to just LED scan.

This one for the scan button is very useful: program can start the LED display control, when the need to scan button, set MODSW = 1, IO scan button, in a very short period of time, after the completion of the scan button, set = MODSW 0, to LED display.

Table 8.14 SEG/COM Mode Control Register

D7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SHARECON	-	-	-	-	-	SHARE.2	SHARE.1	SHARE.0
R/W	-	-	-	-	-	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-	0	0	0

Bit Number	Bit Mnemonic	Description
2	**SHARE.2**	**P1.2 Drive/sink control bit** 0: P1.2 as OP_SEG/IO mode 1: P1.2 as OP_COM/IO mode
1	**SHARE.1**	**P3.0 Drive/sink control bit** 0: P3.0 as OP_SEG/IO mode 1:P3.0 as OP_COM/IO mode
0	**SHARE.0**	**P3.1 Drive/sink control bit** 0: P3.1 as OP_SEG/IO mode 1: P3.1 as OP_COM/IO mode

Table 8.15 LED Clock Control Register

DCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DISPCLK	DCK0.7	DCK0.6	DCK0.5	DCK0.4	DCK0.3	DCK0.2	DCK0.1	DCK0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**DCK0[7:0]**	**LED Clock Selection bit** LED clock frequency = system clock frequency*8*DISPCLK

Table 8.16 COM Scan Width Control Register

DEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DISCOM	DCOM.7	DCOM.6	DCOM.5	DCOM.4	DCOM.3	DCOM.2	DCOM.1	DCOM.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
0 **7-0**	**DCOM[7:0]**	**LED One COM Scan Width Selection bit** One COM Scan Width = LED clock frequency*DISCOM

Note:

T_{LED} is LED clock width, T_{SYS} is the system clock width, T_B is LED scan time width

$T_{LED} = T_{SYS}*8*DISPCLK$

$T_B = (T_{LED}*DISCOM) *S$

S is the number of LED scanning: scanning 4COM, S = 4; S = 5 ,5COM scanning, and so on.

The LED frame 64Hz to display for example, when LED 5COM and RC 16.6MHz as the system clock:

When LEDMODE = 01, LED is displayed as common cathode mode:

If set DISPCLK to 96H (DISPCLK theory, the smaller the better, but also to ensure that TKCOUNT, DISCOM, TKWAIT will not overflow)

$T_{LED} = T_{SYS}*8*DISPCLK$

*= 60.24ns*8*150*

= 72288ns = 72.288us = 0.072288ms

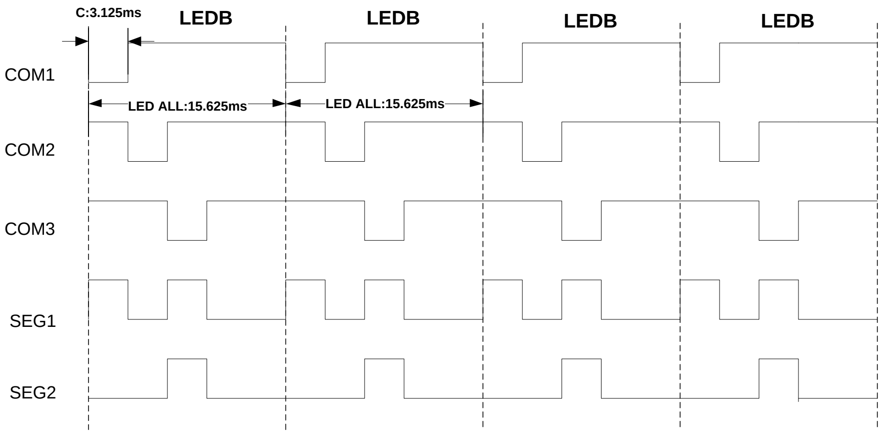
Need to set T_B = 1/64s = 15.625ms

DISCOM = $T_B/S/T_{LED}$ = 15.625ms/5/0.072288 = 43 = 2BH

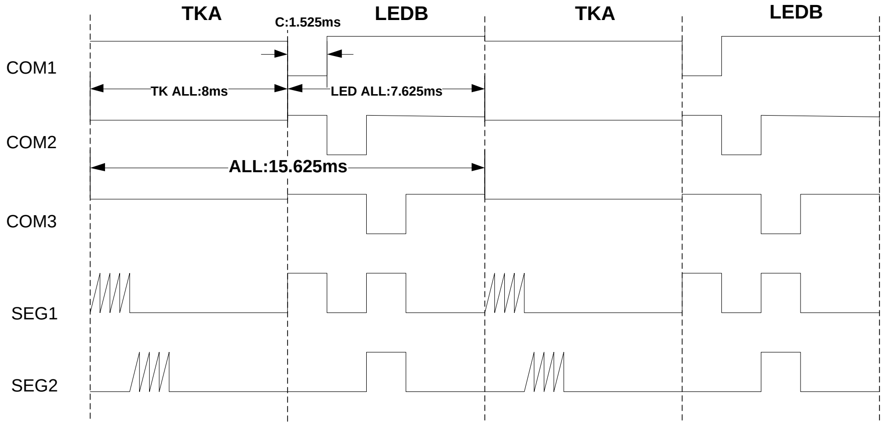
Above are examples of LEDMODE all were negative, model and its algorithm were overcast mode, and a common cathode / anode mode are common, so no longer for example to make narrative .

TK & LED

Tk and LED can achieve the effect of share, the results are as follows:



LED



TK & LED (SHARE)

Table 8.17 SEG Mode Selection Register

DAH-ABH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SEG01 (DAH)	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
SEG02 (DBH)	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SEG[15:0]**	**SEG Mode Selection bit (x = 0-7)** 0: P1.2, P1.7, P2.0-P2.7, P3.1 is I/O 1: P1.2, P1.7, P2.0-P2.7, P3.1 is Segment (LED_S0-LED_S7, LED_S11-LED_S15)

Note: *VCC_IN of SH79F3218 has a maximum flow current limit (see the electrical parameters section), so that more than 8 SEG, please pay attention to evaluate the current flowing through the* V_{DD}.

Table 8.18 COM Mode Selection Register

DDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LEDCOM	-	COM7	COM6	COM5	COM4	COM3	COM2	COM1
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
6-0	**COM[7:1]**	**COM Mode Selection bit (x = 0-6)** 0: P0.7, P1.0-P1.4, P3.1 is I/O 1: P0.7, P1.0-P1.4, P3.1 is COM (LED_C1 - LED_C7)

Note: *GND of SH79F3218 has a maximum flow current limit (see the electrical parameters section), so that more than 4 COM, please pay attention to evaluate the current flowing through the GND.*

LED module does not support single COM mode.

For example:

Common Cathode mode:

SEG01 = 0xAA, SEG02 = 0x50, LEDCOM = 0x55, SEG1, SEG3, SEG5, SEG7, SEG12, SEG14 is SEG pin; COM1, COM3, COM5, COM7 is COM pin;
When LEDRAM, COM1H, COM3H, COM5H, COM7H the corresponding bit is 1, then the corresponding LED is lit;
When LEDRAM, COM1H, COM3H, COM5H, COM7H the corresponding bit is 0, then the corresponding LED goes out.

Common Anode mode:

SEG01 = 0xAA, SEG02 = 0x50, LEDCOM = 0x55, SEG1, SEG3, SEG5, SEG7, SEG12, SEG14 is SEG pin; COM1, COM3, COM5, COM7 is COM pin;
When LEDRAM, COM1L, COM3L, COM5L, COM7L the corresponding bit is 1, then the corresponding LED is lit;
When LEDRAM, COM1L, COM3L, COM5L, COM7L the corresponding bit is 0, then the corresponding LED goes out.

Common Cathode / Anode mode:

SEG01 = 0xAA, SEG02 = 0x50, LEDCOM = 0x55, SEG1, SEG3, SEG5, SEG7, SEG12, SEG14 is SEG pin; COM1, COM3, COM5, COM7 is COM pin;
When LEDRAM, COM1L, COM3L, COM5L, COM7L the corresponding bit is 1, then the corresponding LED is lit; the corresponding bit is 0, then the corresponding LED goes out.
When LEDRAM, COM1H,COM3H, COM5H, COM7H the corresponding bit is 1, corresponding to the common anode LED light; the corresponding bit is 0, corresponding to the common anode led extinguished.

Table 8.19 Brightness Selection Register

DFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LIGHTCOM	-	-	-	-	-	CC3	CC2	CC1
R/W	-	-	-	-	-	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-	0	0	0

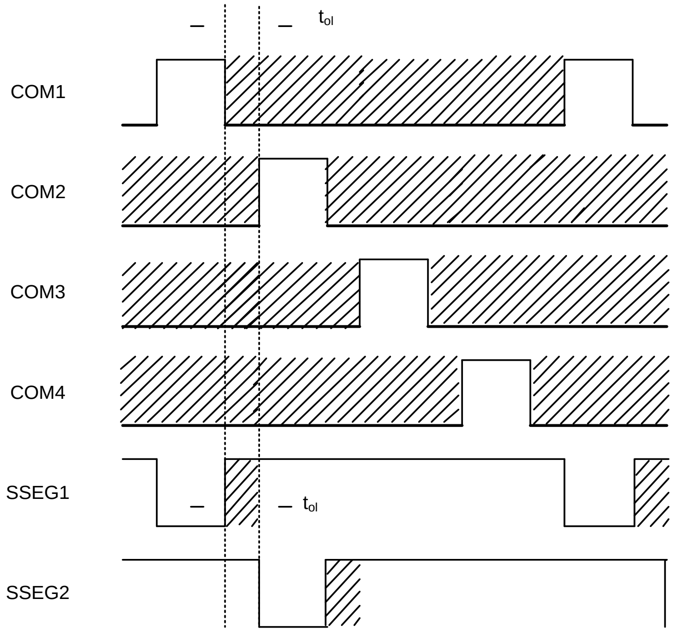
Bit Number	Bit Mnemonic	Description
2-0	**CC[3:1]**	**Brightness Selection bit** 000: COM width 100% 001: COM width 87.5% 010: COM width 75% 011: COM width 62.5% 100: COM width 50% 101: COM width 37.5% 110: COM width 25% 111: COM width 12.5%

8.2.2 Configuration of LED RAM

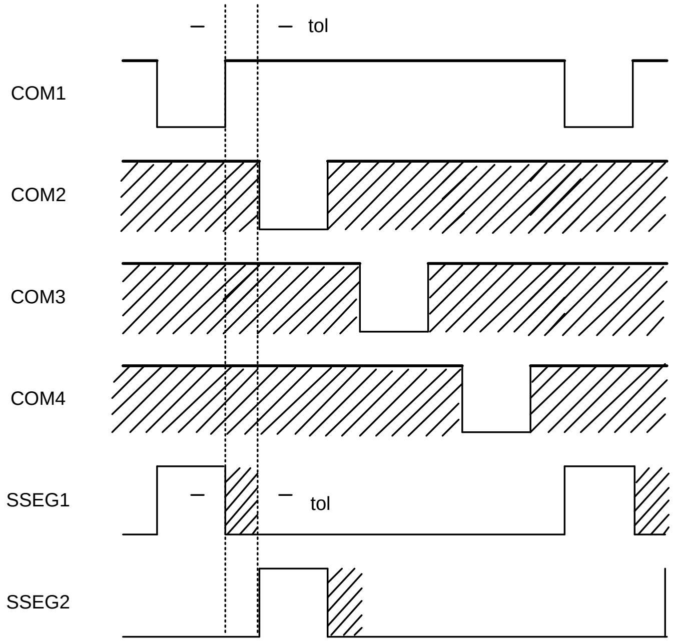
(LED_C1 - C7, LED_S0 - S7, LED_S11 - S15)

Address		7	6	5	4	3	2	1	0
530H	COM1L	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
531H	COM1L	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
532H	COM1H	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
533H	COM1H	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
534H	COM2L	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
535H	COM2L	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
536H	COM2H	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
537H	COM2H	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
538H	COM3L	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
539H	COM3L	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
53AH	COM3H	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
53BH	COM3H	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
53CH	COM4L	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
53DH	COM4L	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
53EH	COM4H	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
53FH	COM4H	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
540H	COM5L	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
541H	COM5L	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
542H	COM5H	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
543H	COM5H	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
544H	COM6L	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
545H	COM6L	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
546H	COM6H	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
547H	COM6H	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
548H	COM7L	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
549H	COM7L	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-
54AH	COM7H	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0
54BH	COM7H	SEG15	SEG14	SEG13	SEG12	SEG11	-	-	-

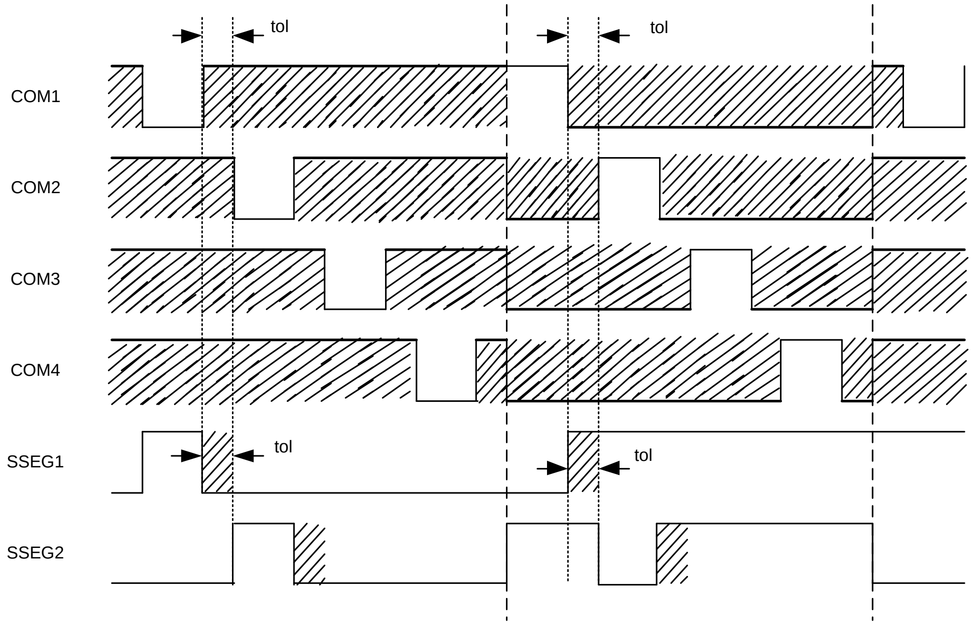
LEDMODE = 00:



LEDMODE = 01:

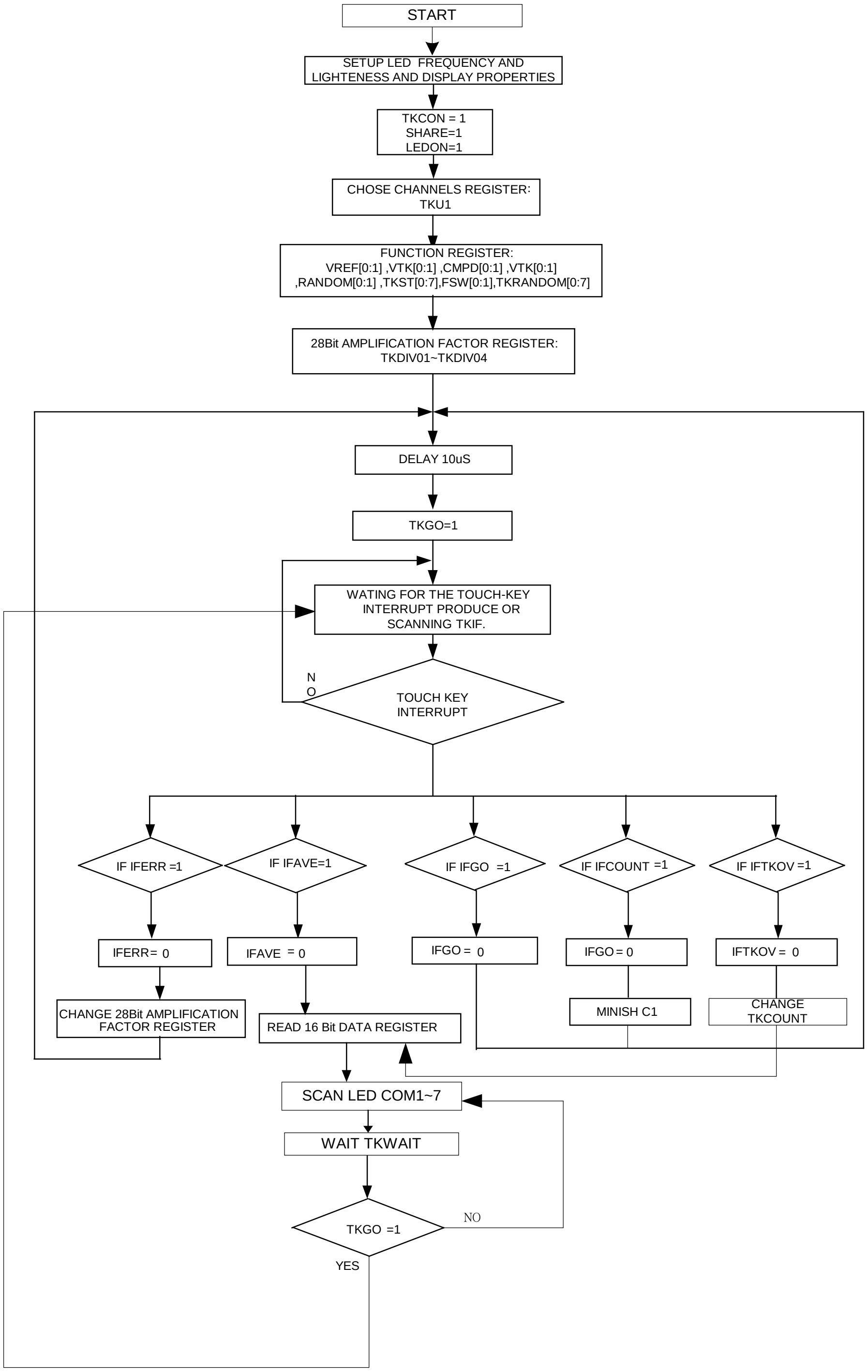


LEDMODE = 1X:



***Note:** t_{OL} is a Common LED signal to the overlap between the time, the value range: 1 Clock LED. The shaded part of the graph is COM floating state.*

8.3 Touch Key Function and LED SHARE Function


START
SETUP LED FREQUENCY AND LIGHTENESS AND DISPLAY PROPERTIES
TKCON = 1
SHARE=1
LEDON=1
CHOSE CHANNELS REGISTER: TKU1
FUNCTION REGISTER: VREF[0:1] ,VTK[0:1] ,CMPD[0:1] ,VTK[0:1] ,RANDOM[0:1] ,TKST[0:7],FSW[0:1],TKRANDOM[0:7]
28Bit AMPLIFICATION FACTOR REGISTER: TKDIV01~TKDIV04
DELAY 10uS
TKGO=1
WATING FOR THE TOUCH-KEY INTERRUPT PRODUCE OR SCANNING TKIF.
NO
TOUCH KEY INTERRUPT
IF IFERR =1
IF IFAVE=1
IF IFGO =1
IF IFCOUNT =1
IF IFTKOV =1
IFERR= 0
IFAVE = 0
IFGO = 0
IFGO = 0
IFTKOV = 0
CHANGE 28Bit AMPLIFICATION FACTOR REGISTER
READ 16 Bit DATA REGISTER
MINISH C1
CHANGE TKCOUNT
SCAN LED COM1~7
WAIT TKWAIT
TKGO =1
NO
YES

Register Flow Diagram

8.3.1 Function Description

SH79F3218 built-in Touch Key function module, which can connect at most 8 keys. When enable LED SHARE function, Touch Key function can connect at most 8 keys. COM1-COM7 is used as LED COM. It is need to notice that, if SHARE function is enable, LED scan function will enable after Touch Key function fininsh data storage. Scan COM1 - COM7, there will be a set of TKWAIT time, if the TKGO is set at the end of the TKWAIT is set to 1, will automatically restart the TK function, start a new round of scan TK and LED.

The startup step of Touch Key:

(1) Set LED: **SEG01-02 = 0FFH, LEDCOM = 7FH, set LED display RAM**

(2) Set scan frequency of LED, set LED brightness and duty

(3) Set **TKCON** bit, enable Touch Key module

(4) Set on-off frequency, reference voltage (Vref), output voltage (OP), key sampling times, key scan sequence, clock width of key scan, enable touch key & LED share function: **FSW[0:1], VREF[0:1], CMPD[0:1], VTK[0;1], TKU1-TKU3, SHARE = 1, LEDON = 1.**

(5) Set 28-bit amplification coefficient register: **DIV01-04**.

(6) Software delay 10uS.

(7) Set **TKGO/**$\overline{\textbf{DONE}}$ bit, start key scan;
Program query interrupt flag bit, **TKGO** is cleared by hardware. Interrupt flag bit: I**FERR**，**IFGO**，**IFAVE**，**IFLED**，**IFTKOV**
If **IFAVE = 1**, reading data register **500H-527H**, program save data result, goto step9;
If **IFERR = 1**, data register arithmetic overflow error, clear **IFERR** and flag bit, reset ting the value of amplification coefficient register, reducing amplification coefficient, go back to step6 to restart scan;
If **IFGO = 1**, key controller start error, clear **IFGO** and flag bit, go back to step7 to restart scan;
If **IFCOUNT = 1**, key scan count overflow error, clear **IFCOUNT** and flag bit, reduce **CX** capacitor or reduce average times.Go back to step7 to restart scan.
Interrupt generate (If **TKIE = 1**), or program query interrupt flag bit, **TKGO** is cleared by hardware.

(8) Touch Key function module:
When scanning Touch Key function is finished, **IFTKOV = 0**, after key clock width time is finished, goto step9;
When scanning Touch Key function is not finished, **IFTKOV = 1**, after scanning key is finished, goto step9.

(9) Enable **LED** scan module: Begin to scan **COM1-COM7**.
Program query register, clear **TKGO/**$\overline{\textbf{DONE}}$ bit, then Touch Key scan is finished, program read key data in RAM to judge.
Program juage whether touch key to judge interrupt flag bit. If no error flag, go back to step7.

(10) LED after the scan is complete, waiting for the **TKWAIT**.

(11) **TKWAIT** time before completion, if **TKGO/**$\overline{\textbf{DONE}}$ position 1, after the completion of the **TKWAIT**, automatic stat T**K** scanning, if **TKGO/**$\overline{\textbf{DONE}}$ bit is 0, touch key not start, but the LED scan and **TKWAIT** time is still circulating until TKGO/$\overline{\text{DONE}}$ position 1, the next starts again.

TOUCHKEY Working Mode Table

TOUCHKEY and LED can be divided into 3 working mode, according to whether SHARE, shown in the following Table:

TKCON	LED_ON	SHARE	Working Mode
1	0	0	TOUCHKEY work independently
1	1	0	TOUCHKEY and LED work separately
1	0	1	TOUCHKEY and LED are in SHARE mode, LED doesn’t work
1	1	1	TOUCHKEY and LED are in SHARE mode, LED work
0	1	X	LED work independently
0	0	X	TOUCHKEY and LED don’t work

SHARE Table of SEG Port

The control signal of SEG port as shown in the following Table:

TKCON	LED_ON	SHARE	TK_STA 0: TK, 1: LED	PXSS	SEGX	PX
1	X	0	X	0	0	I/O
1	X	0	X	0	1	LED
1	X	0	X	1	X	TK
0	X	X	X	0	0	I/O
0	X	X	X	0	1	LED
0	X	X	X	1	X	TK
1	X	1	0	0	X	I/O
1	X	1	0	1	X	TK
1	X	1	1	X	0	I/O
1	X	1	1	X	1	LED

8.4 Pulse Width Modulation（PWM）

8.4.1 Feature

- Two 12-bit PWM module
- Provided overflow and duty interrupt function on every PWM period
- Selectable output polarity

The SH79F3218 has two 12-bit PWM module. The PWM module can provide the pulse width modulation waveform with the period and the duty being controlled, individually. The PWM module clock source and pin output selection are controlled by PWMxCON (x = 1-2) register. The PWM module Period selection are controlled by PWMxPH/L (x = 1-2) register. The PWM module Duty selection are controlled by PWMxDH/L (x = 1-2) register.

8.4.2 Register

Table 8.20 PWM[1:2] control register

E8H, E9H, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1CON (E8H)	PWM1EN	PWM1S	PWM1CK1	PWM1CK0	-	PWM1IE	PWM1IF	PWM1SS
PWM2CON (E9H)	PWM2EN	PWM2S	PWM2CK1	PWM2CK0	-	PWM2IE	PWM2IF	PWM2SS
R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	-	0	0	0

Bit Number	Bit Mnemonic	Description
7	**PWMxEN**	**PWMx module enable control bit** 0: disable PWMx module 1: enable PWMx module
6	**PWMxS**	**PWMx output normal mode of duty cycle** 0: high active, PWMx output high during duty time, output low during remain period time 1: low active, PWMx output low during duty time, output high during remain period time
5-4	**PWMxCK[1:0]**	**PWMx clock select bit** 00: system clock/1 01: system clock /2 10: system clock /4 11: system clock /8
2	**PWMxIE**	**PWMx interrupt enable bit** 0: Disable PWMx interrupt 1: Enable PWMx interrupt
1	**PWMxIF**	**PWMx interrupt flag bit** 0: Clear by software 1: Set by hardware to indicate that the PWMx period counter overflow
0	**PWMxSS**	**PWMx output control bit** 0: PWMx is disable, shared as I/O 1: PWMx is enable

Note: *when PWMEN is cleared 0, the PWM output is immediately shut down.*

Table 8.21 PWM1 Period Control Register (PWM1PH/L)

EAH, EBH, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1PH (EAH)	-	-	-	-	PWM1P.11	PWM1P.10	PWM1P.9	PWM1P.8
PWM1PL (EBH)	PWM1P.7	PWM1P.6	PWM1P.5	PWM1P.4	PWM1P.3	PWM1P.2	PWM1P.1	PWM1P.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
11-0	**PWM1P[11:0]**	**PWM1output Period = PWM1P * PWM clock** When PWM1P = 00H, if PWM1S = 0, PWM1 output low When PWM1P = 00H, if PWM1S = 1, PWM1 output high

Table 8.22 PWM2 Period Control Register (PWM2PH/L)

ECH, EDH, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2PH (ECH)	-	-	-	-	PWM2P.11	PWM2P.10	PWM2P.9	PWM2P.8
PWM2PL (EDH)	PWM2P.7	PWM2P.6	PWM2P.5	PWM2P.4	PWM2P.3	PWM2P.2	PWM2P.1	PWM2P.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
11-0	**PWM2P[11:0]**	**PWM2output Period = PWM2P * PWM clock** When PWM2P = 00H, if PWM2S = 0, PWM2 output low When PWM2P = 00H, if PWM2S = 1, PWM2 output high

***Note:** Modifying the register PWMxPH will make the output of the PWMx come into effect in the next cycle. Users need to modify the PWMxPL, and then modify the PWMxPH to modify the PWM cycle.*

Table 8.23 PWM1 Duty Control Register (PWM1DH/L)

E4H, E5H, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM1DH (E4H)	-	-	-	-	PWM1D.11	PWM1D.10	PWM1D.9	PWM1D.8
PWM1DL (E5H)	PWM1D.7	PWM1D.6	PWM1D.5	PWM1D.4	PWM1D.3	PWM1D.2	PWM1D.1	PWM1D.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
11-0	**PWM1D[11:0]**	**PWM1 Duty control bit** **Special Case:** 1 When PWM1P ≤ PWM1D: If PWM1S = 0, PWM1 output high level If PWM1S = 1, PWM1 output low level 2 WhenPWM1D = 00H: If PWM1S = 0, PWM1 output low level If PWM1S = 1, PWM1 output high level

Table 8.24 PWM2 Duty Control Register PWM2DH/L

E6H, E7H, Bank1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM2DH (E6H)	-	-	-	-	PWM2D.11	PWM2D.10	PWM2D.9	PWM2D.8
PWM2DL (E7H)	PWM2D.7	PWM2D.6	PWM2D.5	PWM2D.4	PWM2D.3	PWM2D.2	PWM2D.1	PWM2D.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
11-0	**PWM2D[11:0]**	**PWM2 Duty control bit** **Special Case:** 1 When PWM2P ≤ PWM2D: If PWM2S = 0, PWM2 output high level If PWM2S = 1, PWM2 output low level 2 WhenPWM2D = 00H: If PWM2S = 0, PWM2 output low level If PWM2S = 1, PWM2 output high level

Note: *the modified register PWMxDH will make the output of the PWMx come into effect in the next cycle. Users need to modify the PWMxDL, next to modify the PWMxDH, and then change the PWM duty cycle.*

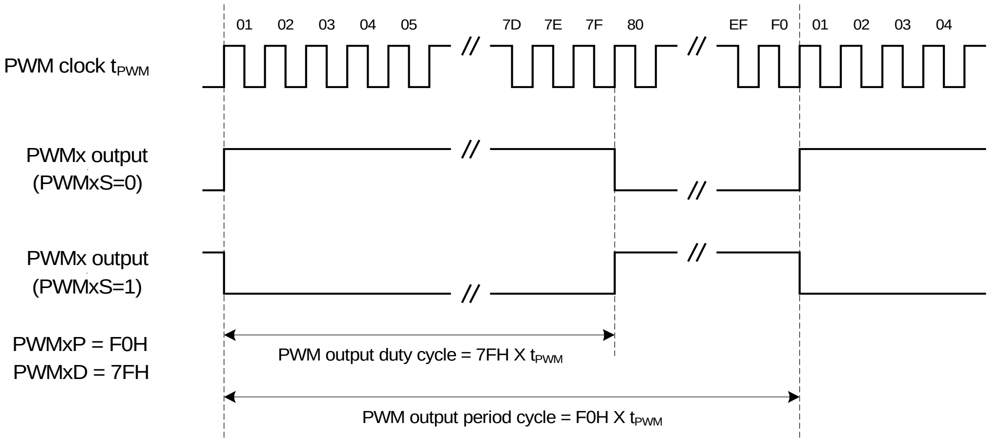
Note:

(1) PWM1EN bit control PWMx disable or enable

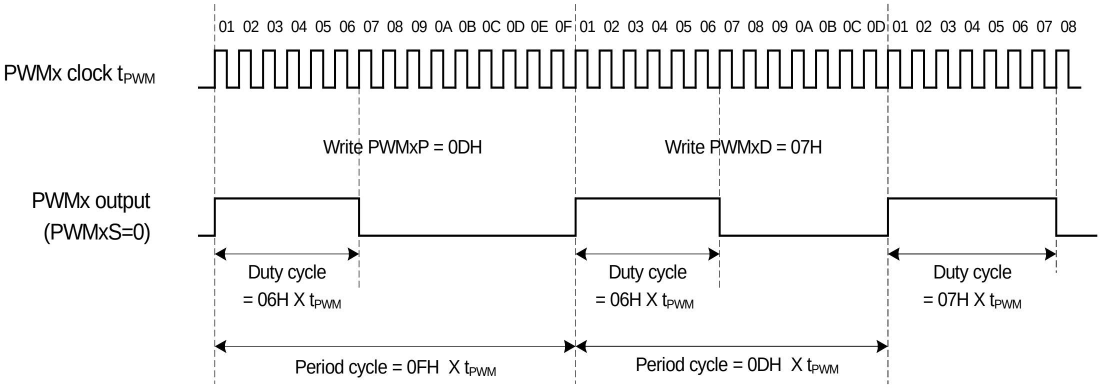
(2) PWMxSS(x = 1-2) bit control Port is IO or PWMx output port

(3) EPWMx bit in IEN1 register and PWMxIE bit in PWMxCON are control PWMx interrupt disable or enable

(4) If PWMxEN = 1, PWMx module enable, but PWMxSS(x = 1-2) = 0, PWMx output disable, PWMx can be used as a 12-bit timer. If the interrupt control register EPWMx IEN1 position 1 and PWMxIF = 1, then the PWMx interrupt occurs.


01 02 03 04 05 // 7D 7E 7F 80 // EF F0 01 02 03 04
PWM clock t_{PWM}
PWMx output
(PWMxS=0)
PWMx output
(PWMxS=1)
PWMxP = F0H
PWMxD = 7FH
PWM output duty cycle = 7FH X t_{PWM}
PWM output period cycle = F0H X t_{PWM}

PWMx output example


01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 01 02 03 04 05 06 07 08
PWMx clock t_{PWM}
Write PWMxP = 0DH
Write PWMxD = 07H
PWMx output
(PWMxS=0)
Duty cycle
= 06H X t_{PWM}
Duty cycle
= 06H X t_{PWM}
Duty cycle
= 07H X t_{PWM}
Period cycle = 0FH X t_{PWM}
Period cycle = 0DH X t_{PWM}

PWMx output Period or Duty cycle changing example

8.5 EUART

8.5.1 Features

- The SH79F3218 has one enhanced EUART
- The baud rate generator is an 15 bit up-counting timer
- Enhancements over the standard 8051 the EUART include Framing Error detection and automatic address recognition
- The EUART can be operated in four modes

8.5.2 EUART Mode Description

The EUART can be operated in 4 modes. Users must initialize the SCON before any communication can take place. This involves selection of the Mode and the baud rate.

In all of the 4 modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. This will generate a clock on the TxD pin and shift in 8 bits on the RxD pin. Reception is initiated in the other modes by the incoming start bit if RI = 0 and REN = 1. The external transmitter will start the communication by transmitting the start bit.

EUART Mode Summary

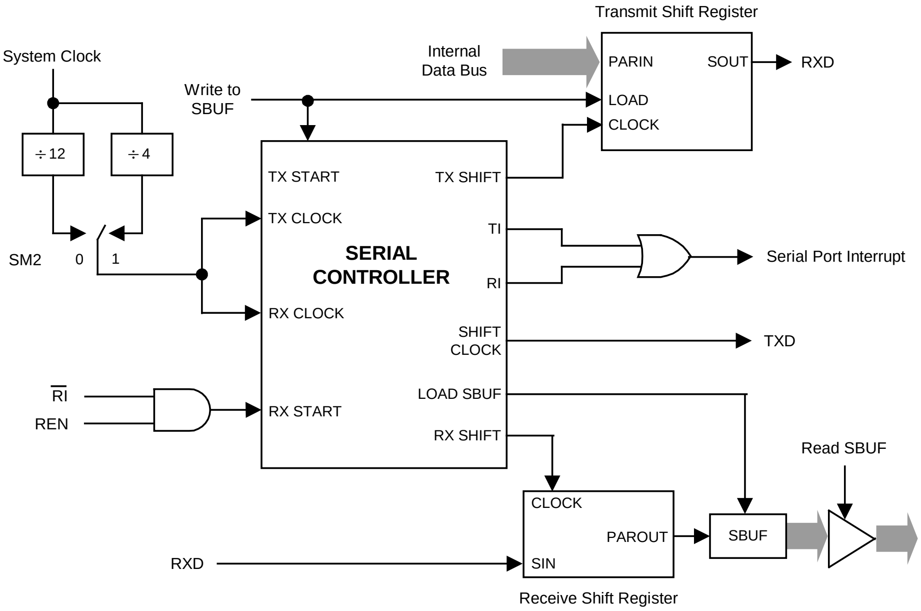
SM0	SM1	Mode	Type	Baud Clock	Frame Size	Start Bit	Stop Bit	9th bit
0	0	0	Synch	f_{SYS}/(4 or 12)	8 bits	NO	NO	None
0	1	1	Asynch	Own baud-rate generator overflow rate/16	10 bits	1	1	None
1	0	2	Asynch	f_{SYS}/(32 or 64)	11 bits	1	1	0, 1
1	1	3	Asynch	Own baud-rate generator overflow rate/16	11 bits	1	1	0, 1

Mode0: Synchronous Mode, Half duplex

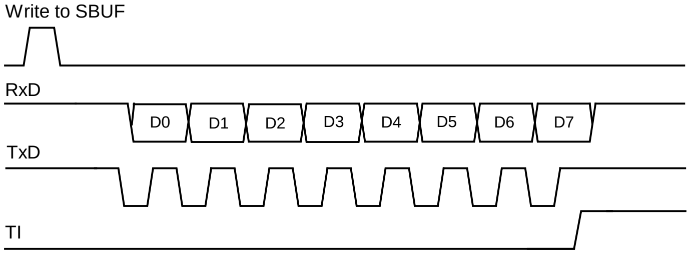
This mode provides synchronous communication with external devices. In this mode serial data is transmitted and received on the RxD line. TxD is used to output the shift clock. The TxD clock is provided by the SH79F3218 whether the device is transmitting or receiving. This mode is therefore a half duplex mode of serial communication. In this mode, 8 bits are transmitted or received per frame. The LSB is transmitted/received first.

The baud rate is programmable to either 1/12 or 1/4 of the system clock. This baud rate is determined in the SM2 bit (SCON.5). When this bit is set to 0, the serial port runs at 1/12 of the system clock. When set to 1, the serial port runs at 1/4 of the system clock. The only difference from standard 8051 is that SH79F1618 in the mode 0 has variable baud rate.

The functional block diagram is shown below. Data enters and exits the serial port on the RxD line. The TxD line is used to output the SHIFT CLOCK.

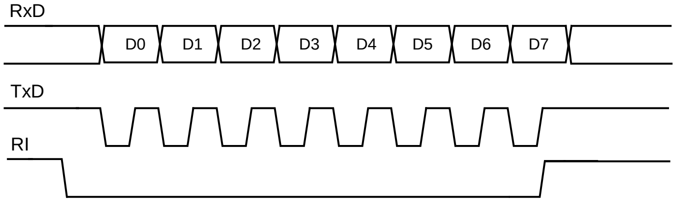


Any instruction that uses SBUF as a destination register ("write to SBUF" signal) will start the transmission. The next system clock tells the Tx control block to commence a transmission. The data shift occurs at the falling edge of the SHIFT CLOCK, and the contents of the transmit shift register is shifted one position from left to the right. As data bits shift to the right, zeros come in from the left. After transmission of all 8 bits in the transmit shift register, the Tx control block will deactivates SEND and sets TI (SCON.1) at the rising edge of the next system clock.



Send Timing of Mode 0

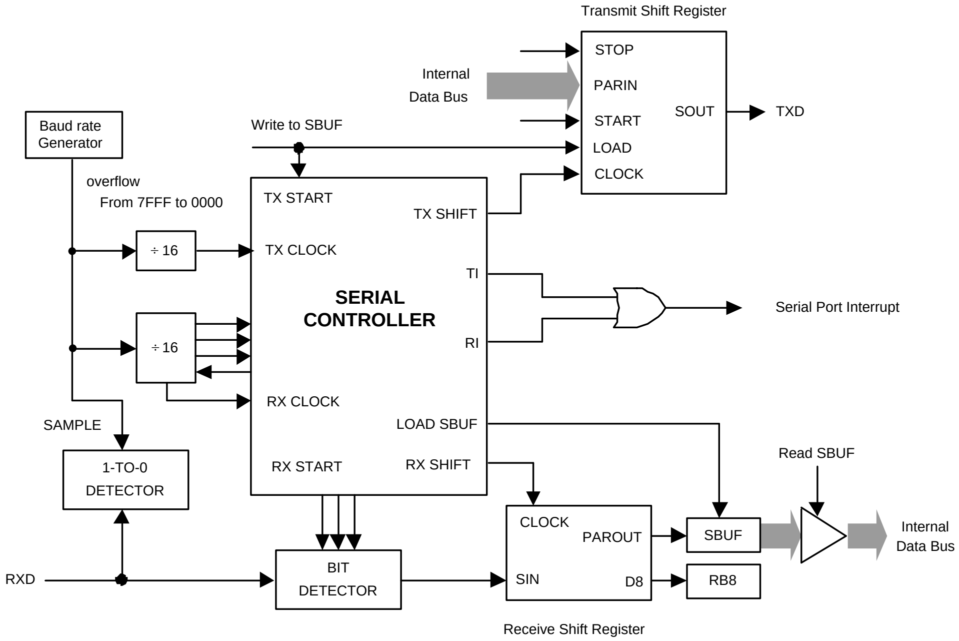
Reception is initiated by the condition REN (SCON.4) = 1 and RI (SCON.0) = 0. The next system clock activates RECEIVE. The data latch occurs at the rising edge of the SHIFT CLOCK, and the contents of the receive shift register are shifted one position to the left. After the receiving of all 8 bits into the receive shift register, the RX control block will deactivates RECEIVE and sets RI at the rising edge of the next system clock, and the reception will not be enabled till the RI is cleared by software.



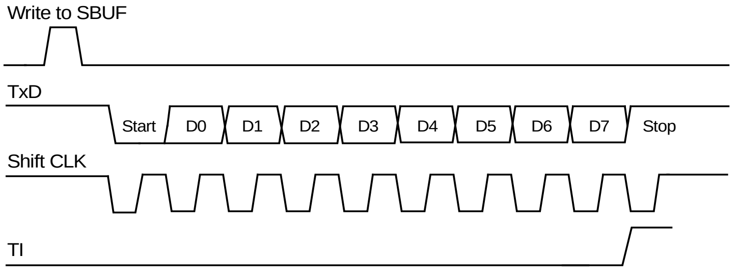
Receive Timing of Mode 0

Mode1: 8-Bit EUART, Variable Baud Rate, Asynchronous Full-Duplex

This mode provides the 10 bits full duplex asynchronous communication. The 10 bits consist of a start bit (logical 0), 8 data bits (LSB first), and a stop bit (logical 1). When receiving, the eight data bits are stored in SBUF and the stop bit goes into RB8 (SCON.2). The serial receive and transmit baud rate is 1/16 of the Timer4/2 overflow (Refer to **Baud Rate** Section for details). The functional block diagram is shown below.



Transmission begins with a "write to SBUF" signal, and it actually commences at the next system clock following the next rollover in the divide-by-16 counter (divide baud-rate by 16), thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SUBF" signal. The start bit is firstly put out on TxD pin, then are the 8 bits of data. After all 8 bits of data in the transmit shift register are transmitted, the stop bit is put out on the TxD pin, and the TI flag is set at the same time that the stop is send.



Send Timing of Mode 1

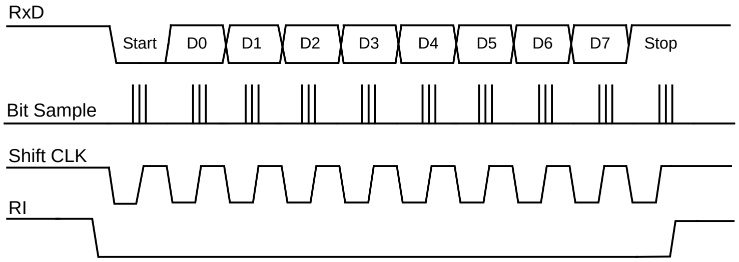
Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. This helps the divide-by-16 counter to synchronize with the serial datas of RXD pin. The divide-by-16 counter divides each bit time into 16 states. The bit detector samples the value of RxD at the 7th, 8th and 9th counter states of each bit time. At least 2 the sampling values have no difference in the state of the three samples, data can be received This is done for noise rejection. If the first bit after the falling edge of RxD pin is not 0, which indicates an invalid start bit, and the reception is immediately aborted. The receive circuits are reset and again waiting for a falling edge in the RxD pin. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the shift register. After shifting in 8 data bits and the stop bit, the SBUF and RB8 are loaded and RI are set, if the following conditions are met:

(1) RI must be 0

(2) Either SM2 = 0, or the received stop bit = 1

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost.

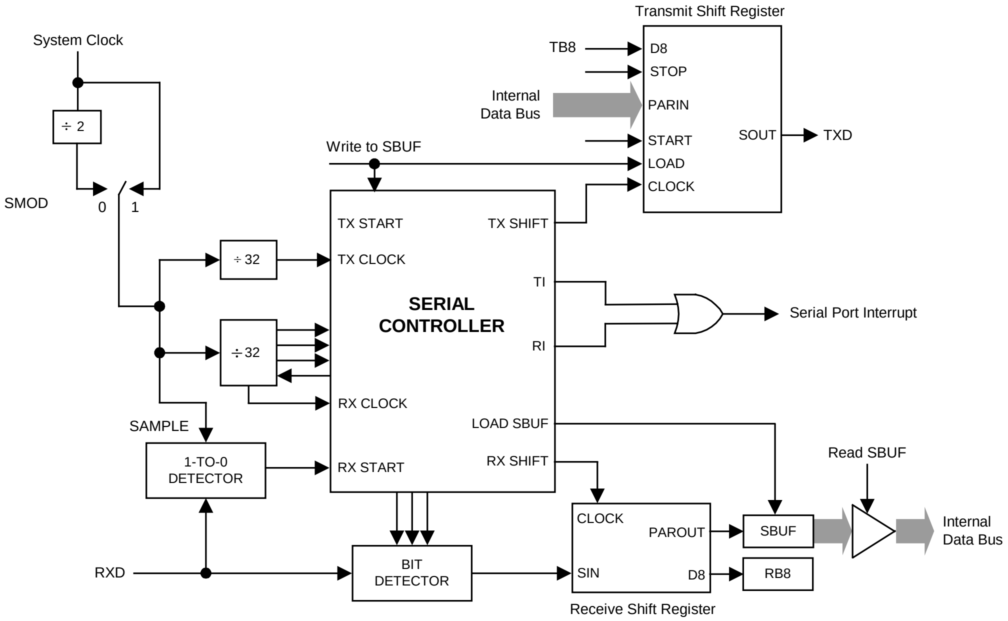
At the time, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



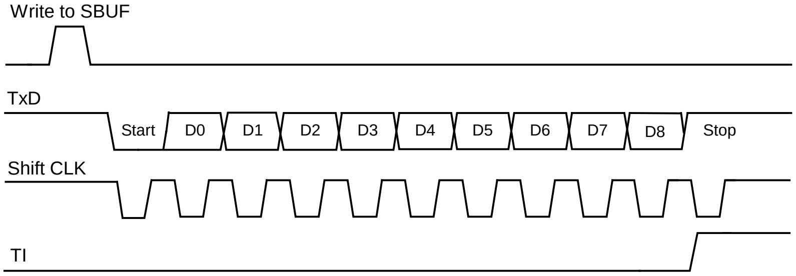
Receive Timing of Mode 1

Mode2: 9-Bit EUART, Fixed Baud Rate, Asynchronous Full-Duplex

This mode provides the 11 bits full duplex asynchronous communication. The 11 bit consists of one start bit (logical 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logical 1). Mode 2 supports multiprocessor communications and hardware address recognition (Refer to Multiprocessor Communication Section for details). When data is transmitted, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1, for example, the parity bit P in the PSW or used as data/address flag in multiprocessor communications. When data is received, the 9th data bit goes into RB8 and the stop bit is not saved. The baud rate is programmable to either 1/32 or 1/64 of the system working frequency, as determined by the SMOD bit in PCON. The functional block diagram is shown below:



Transmission begins with a “write to SBUF” signal, the “write to SBUF” signal also loads TB8 into the 9th bit position of the transmit shift register. Transmission actually commences at the next system clock following the next rollover in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the “write to SUBF” signal). The start bit is firstly put out on TxD pin, then are the 9 bits of data. After all 9 bits of data in the transmit shift register are transmitted, the stop bit is put out on the TxD pin, and the TI flag is set at the same time, this will be at the 11th rollover of the divide-by-16 counter after a write to SBUF.



Send Timing of Mode 2

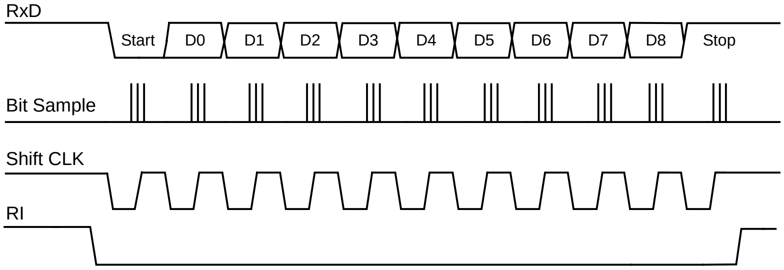
Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide-by-16 counter. The 16 states of the counter divide each bit time into 16ths. The bit detector samples the value of RxD at the 7th, 8th and 9th counter state of each bit time. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the first bit detected after the falling edge of RxD pin is not 0, which indicates an invalid start bit, and the reception is immediately aborted. The receive circuits are reset and again looks for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the shift register. After shifting in 9 data bits and the stop bit, the SBUF and RB8 are loaded and RI is set if the following conditions are met:

(1) RI must be 0

(2) Either SM2 = 0, or the received 9th bit = 1 and the received byte accords with Given Address

If these conditions are met, then the 9th bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost.

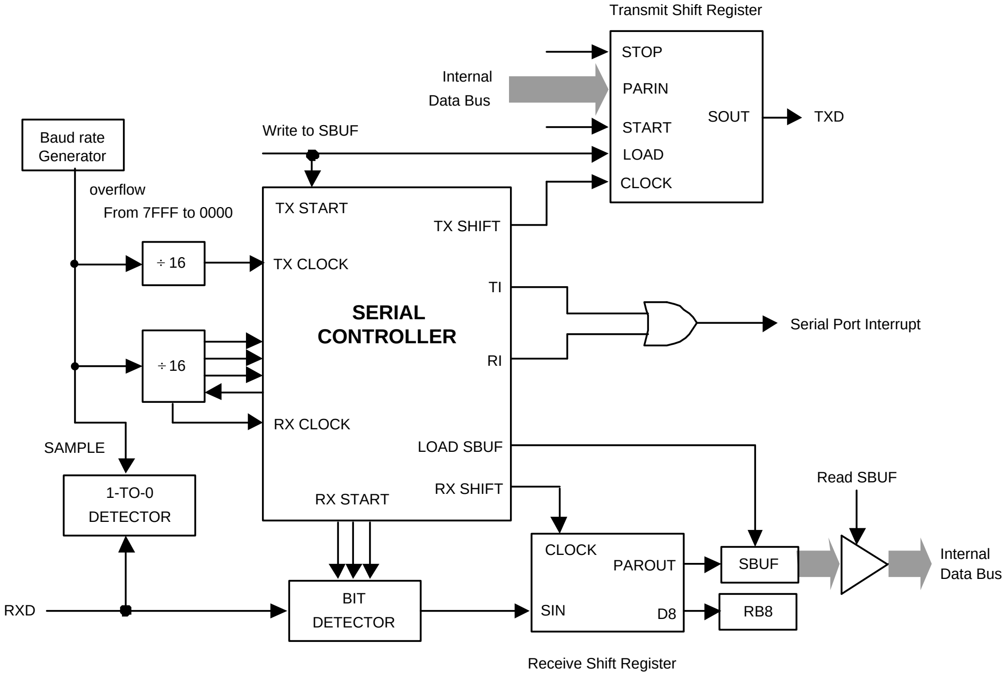
At the time, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



Receive Timing of Mode 2

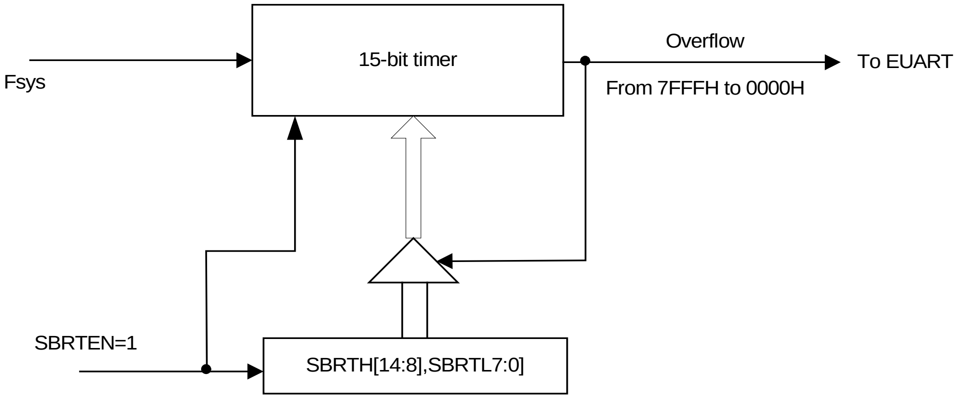
Mode3: 9-Bit EUART, Variable Baud Rate, Asynchronous Full-Duplex

Mode3 uses transmission protocol of the Mode2 and baud rate generation of the Mode1.



8.5.3 Baud Rate Generate

The baud rate generator is an 15 bit up-counting timer.



Baudrate Generator for EUART

$$\text{BaudRate} = 2^{\text{SMOD}} \times (\frac{f_{\text{SYS}}}{64})$$

$$\text{SBRToverflowrate} = \frac{\text{Fsys}}{32768 - \text{SBRT}}, \quad SBRT = [SBRTH, SBRTL]$$

So that, the baud rate computational formula of EUART in different modes as shown below.

Mode0:

In Mode0, the baud rate is programmable to either 1/12 or 1/4 of the system frequency. This baud rate is determined by SM2 bit. When set to 0, the serial port runs at 1/12 of the system clock. When set to 1, the serial port runs at 1/4 of the system clock.

Mode1 and Mode3:

In Mode1 & Mode3, the baud rate can be fine adjusted.

The Mode1 & 3 baud rate equations are shown below,

$$BaudRate = \frac{Fsys}{16 \times (32768 - SBRT) + BFINE}$$

For example: Fsys = 8MHz, to get 115200Hz baud rate, computing method of SBRT and SFINE as shown below:

8000000/16/115200 = 4.34

SBRT = 32768 - 4 = 32764

115200 = 8000000/(16 X 4 + BFINE)

BFINE = 5.4 $\approx$ 5

This fine tuning method to calculate the actual baud rate is 115942Hz and the error is 0.64%, but the error is 8.5% In the past computing method.

In Mode2, the baud rate is programmable to either 1/32 or 1/64 of the system clock. This baud rate is determined by the SMOD bit (PCON.7). When this bit is set to 0, the serial port runs at 1/64 of the clock. When set to 1, the serial port runs at 1/32 of the clock.

$$\text{BaudRate} = 2^{\text{SMOD}} \times (\frac{f_{\text{SYS}}}{64})$$

8.5.4 Automatic (Hardware) Address Recognition

In Mode2 & 3, setting the SM2 bit will configure EUART act as following: when a stop bit is received, EUART will generate an interrupt only if the 9th bit that goes into RB8 is logic 1 (address byte) and the received data byte matches the EUART slave address. Following the received address interrupt, the slave should clear its SM2 bit to enable interrupts on the reception of the following data byte(s).

The 9-bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. When the master processor wants to transmit a block of data to one of the slaves, it first sends out the address of the targeted slave (or slaves). All the slave processors should have their SM2 bit set high when waiting for an address byte, which ensures that they will be interrupted only by the reception of an address byte. The Automatic address recognition feature further ensures that only the addressed slave will be interrupted. The address comparison is done by hardware not software.

After being interrupted, the addressed slave clears the SM2 bit to receive data bytes. The un-addressed slaves will be unaffected, as they will be still waiting for their address. Once the entire message is received, the addressed slave should set its SM2 bit to ignore all transmissions until it receives the next address byte.

The Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given Address. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. The slave address is an 8-bit value specified in the SADDR register. The SADEN register is actually a mask for the byte value in SADDR. If a bit position in SADEN is 0, then the corresponding bit position in SADDR is don't care. Only those bit positions in SADDR whose corresponding bits in SADEN are 1 are used to obtain the Given Address. This gives the user flexibility to address multiple slaves without changing the slave address in SADDR. Use of the Given Address allows multiple slaves to be recognized while excluding others.

	Slave 1	**Slave 2**
SADDR	10100100	10100111
SADEN (0 mask)	11111010	11111001
Given Address	10100x0x	10100xx1
Broadcast Address (OR)	1111111x	11111111

The Given address for slave 1 and 2 differ in the LSB. For slave 1, it is a don't care, while for slave 2 it is 1. Thus to communicate only with slave 1, the master must send an address with LSB = 0 (10100000). Similarly the bit 1 is 0 for slave 1 and don't care for slave 2. Hence to communicate only with slave 2 the master has to transmit an address with bit 1 = 1 (1010 0011). If the master wishes to communicate with both slaves simultaneously, then the address must have bit 0 = 1 and bit 1 = 0. The bit 2 position is don't care for both the slaves. This allows two different addresses to select both slaves (1010 0001 and 1010 0101).

The master can communicate with all the slaves simultaneously with the Broadcast Address. This address is formed from the logical OR of the SADDR and SADEN. The zeros in the result are defined as don't cares. In most cases, the Broadcast Address is FFh, this address will be acknowledged by all slaves.

On reset, the SADDR and SADEN are initialized to 00h. This results in Given Address and Broadcast Address being set as XXXXXXXX (all bits don't care). This effectively removes the multiprocessor communications feature, since any selectivity is disabled. This ensures that the EUART will reply to any address, which it is backwards compatible with the 80C51 microcontrollers that do not support automatic address recognition. So the user may implement multiprocessor by software address recognition mentioned above.

8.5.5 Error Detection

Error detection is available when the SSTAT bit in register PCON is set to logic 1.The SSTAT bit must be logic 1 to access any of the status bits (FE, RXOV, and TXCOL). The SSTAT bit must be logic 0 to access the Mode Select bits (SM0, SM1, and SM2).All the 3 bits should be cleared by software after they are set, even when the following frames received without any error will not be cleared automatically.

Transmit Collision

The Transmit Collision bit (TXCOL bit in register SCON) reads ‘1’ if RI is set 0 and user software writes data to the SBUF register while a transmission is still in progress. If this occurs, the new data will be ignored and the transmit buffer will not be written.

Receive Overrun

The Receive Overrun bit (RXOV in register SCON) reads ‘1’ if a new data byte is latched into the receive buffer before software has read the previous byte. The previous data is lost when this happen.

Frame Error

The Frame Error bit (FE in register SCON) reads ‘1’ if an invalid (low) STOP bit is detected.

Note: *TXD pin must be set as output high level before sending*.

8.5.6 Register

Table 8.25 Power Control Register

87H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD	SSTAT	-	-	GF1	GF0	PD	IDL
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**SMOD**	**Baud rate doubler** 0: If set in Mode2, the baud-rate of EUART is system clock/64 1: If set in Mode2, the baud-rate of EUART is system clock/32
6	**SSTAT**	**SCON [7:5] function select bit** 0: SCON [7:5] operates as SM0, SM1, SM2 1: SCON [7:5] operates as FE, RXOV, TXCOL
3-2	**GF[1:0]**	**General purpose flags for software use**
1	**PD**	**Power-Down mode control bit**
0	**IDL**	**Idle mode control bit**

Table 8.26 EUART Control & Status Register

98H, Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON	SM0 /FE	SM1 /RXOV	SM2 /TXCOL	REN	TB8	RB8	TI	RI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-6	**SM[0:1]**	**EUART Serial mode control bit, when SSTAT = 0** 00: mode 0, Synchronous Mode, fixed baud rate 01: mode 1, 8 bit Asynchronous Mode, variable baud rate 10: mode 2, 9 bit Asynchronous Mode, fixed baud rate 11: mode 3, 9 bit Asynchronous Mode, variable baud rate
7	**FE**	**EUART Frame Error flag, when FE bit is read, SSTAT bit must be set 1** 0: No Frame Error, clear by software 1: Frame error occurs, set by hardware
6	**RXOV**	**EUART Receive Over flag, when RXOV bit is read, SSTAT bit must be set 1** 0: No Receive Over, clear by software 1: Receive over occurs, set by hardware
5	**SM2**	**EUART Multi-processor communication enable bit (9th bit '1' checker), when SSTAT = 0** 0: In Mode0, baud-rate is 1/12 of system clock In Mode1, disable stop bit validation check, any stop bit will set RI to generate interrupt In Mode2 & 3, any byte will set RI to generate interrupt 1: In Mode0, baud-rate is 1/4 of system clock In Mode1, Enable stop bit validation check, only valid stop bit (1) will set RI to generate interrupt In Mode2 & 3, only address byte (9th bit = 1) will set RI to generate interrupt
5	**TXCOL**	**EUART Transmit Collision flag, when TXCOL bit is read, SSTAT bit must be set 1** 0: No Transmit Collision, clear by software 1: Transmit Collision occurs, set by hardware
4	**REN**	**EUART Receiver enable bit** 0: Receive Disable 1: Receive Enable
3	**TB8**	**The 9th bit to be transmitted in Mode2 & 3 of EUART, set or clear by software**
2	**RB8**	**The 9th bit to be received in Mode1, 2 & 3 of EUART** In Mode0, RB8 is not used In Mode1, if receive interrupt occurs, RB8 is the stop bit that was received In Modes2 & 3 it is the 9th bit that was received
1	**TI**	**Transmit interrupt flag of EUART** 0: cleared by software 1: Set by hardware at the end of the 8th bit time in Mode0, or at the beginning of the stop bit in other modes
0	**RI**	**Receive interrupt flag of EUART** 0: cleared by software0 1: Set by hardware at the end of the 8th bit time in Mode0, or during the stop bit time in other modes

Table 8.27 EUART Data Buffer Register

99H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUF	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SBUF[7:0]**	This SFR accesses two registers; a transmit shift register and a receive latch register A write of SBUF will send the byte to the transmit shift register and then initiate a transmission A read of SBUF returns the contents of the receive latch

Table 8.28 EUART Slave Address & Address Mask Register

9AH-9BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SADDR (9AH)	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
SADEN (9BH)	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SADDR[7:0]**	**SFR SADDR defines the EUART's slave address**
7-0	**SADEN[7:0]**	**SFR SADEN is a bit mask to determine which bits of SADDR are checked against a received address** 0: Corresponding bit in SADDR is a "don't care" 1: Corresponding bit in SADDR is checked against a received address

Table 8.29 EUART Baudrate generator Register

C7H，BFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBRTH (C7H)	SBRTEN	SBRT.14	SBRT.13	SBRT.12	SBRT.11	SBRT.10	SBRT.9	SBRT.8
SBRTL (BFH)	SBRT.7	SBRT.6	SBRT.5	SBRT.4	SBRT.3	SBRT.2	SBRT.1	SBRT.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**SBRTEN**	**EUART Baudrate generator control bit** 0: disable(default) 1: enable
6-0 **7-0**	**SBRT[14:0]**	**EUART Baudrate generator data**

Table 8.30 EUART Baudrate generator Bfine Register

BEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SFINE	-	-	-	-	SFINE.3	SFINE.2	SFINE.1	SFINE.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	**SFINE[3:0]**	**EUART Baudrate generator Bfine data Register**

8.6 TWI

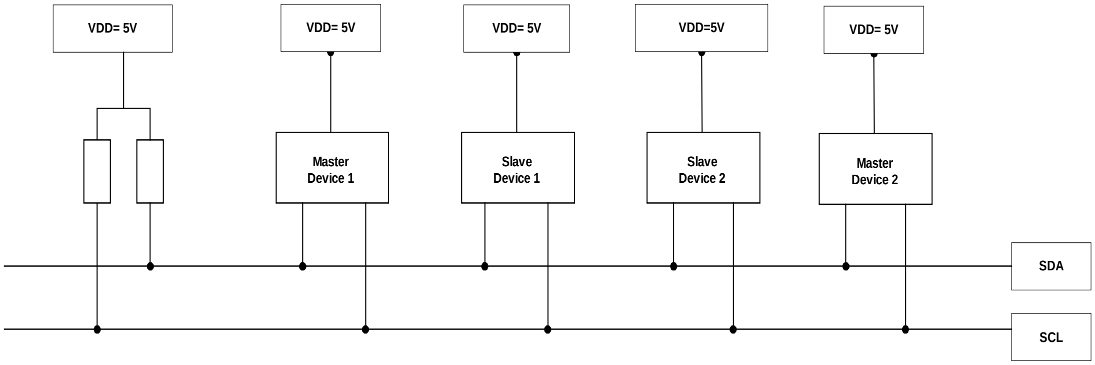
8.6.1 Features

- Two Wire Interface, simple and fast
- Master and Slave operation Supported
- Device are operated as Transmitter or Receiver
- Multi-master Arbitration Supported
- TWI Timeout Detection
- Wake-up system when SH79F3218 is in IDLE Mode
- Programable address

TWI serial bus adopt two wires (SDA and SCL) to transmit messages between bus and device. SH79F3218 is totally in conformity with TWI bus standard, transmitting and processing bytes automatically, and tracking serial communication.

TWI function need 16.6 MHz system frequency, when system period is 128K/32.768KHz, OSC2 16.6MRC can't disable to advoid that TWI can't communicate.

The following diagram shows a typical TWI configuration using one Master controller and many Slave peripherals.The protocol allows the system to interconnect up to 128 different devices using only two lines.



8.6.2 Data Transformat

Data Transformat

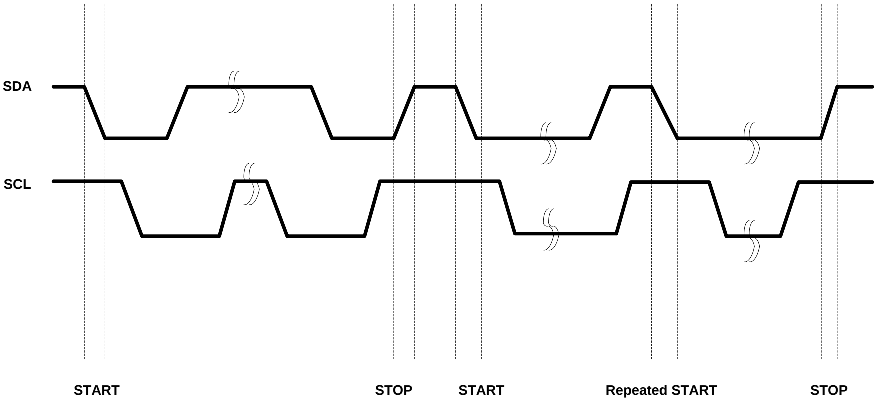
Each data bit transferred on the TWI data transfer lines is accompanied by a pulse on the clock line. The level of the data line must be stable when the clock line is high. The only exception to this rule is for generating START and STOP conditions.

As with I2C, TWI defines two special waveform: START and STOP condition. A high to low transition of SDA line while SCL is high defines as START condition; A low to high transition of SDA line while SCL is high defines as STOP condition. START and STOP conditions are always generated by the bus master.

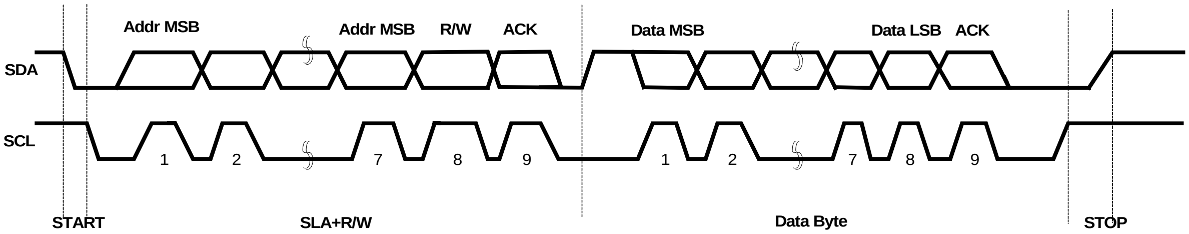
The Master can initiate and terminate a data transmission. The transmission is initiated when the Master transfers a START condition, and it is terminated when the Master transfers a STOP condition. Between START and STOP condition, the bus is considered as busy. The other masters shouldn't try to initiate a transfer. In Busy mode, if the Master initiates START condition again, it will be defined as REPEATED START condition to indicate that the Master wishes to initiate a new transfer without relinquishing bus. After a REPEATED START, the bus will be still in Busy mode until the next STOP. Considering that the features of REPEATED START condition and START condition are same, except for special statement, START condition will be used to describe both START and REPEATED START conditions for the remainder of this datasheet.

All data packets (including address packets) are 9-bit long, consisting of one data byte and an acknowledge bit. During a data transfer, the Master generates the clock and the START and STOP conditions, while the Receiver is responsible for acknowledging the reception. An Acknowledge (ACK) is signaled by the Receiver through pulling the SDA line low during the ninth SCL pulse. If the Receiver hold high at the ninth SCL pulse, a NACK is signaled. When the Receiver has received the last byte, or for some reason cannot receive any more bytes, it should respond NACK signal. The MSB of the data byte is transmitted first.

A transmission basically consists of a START condition, a SLA + R/W, one or more data packets and a STOP condition.An empty message, consisting of a START followed by a STOP condition, is illegal. Note that the wired-AND can be used to implement handshaking between the Master and the Slave. The Slave can extend the SCL low period by pulling the SCL line low. This is useful if the clock speed set up by the Master is too fast for the Slave, or the Slave needs extra time for processing between the data transmissions. The Slave extending the SCL low period will not affect the SCL high period, which is determined by the Master. As a consequence, the Slave can reduce TWI data transfer speed by prolonging the TWI duty cycle.



When generating ACK signal, SH79F3218 will pull the SDA line low. During setting interrupt flag bit, SH79F3218 pull the SCL line low, releasing the SDA line. Clearing TWINT flag after interrupt process is finished, releasing the SCL line.

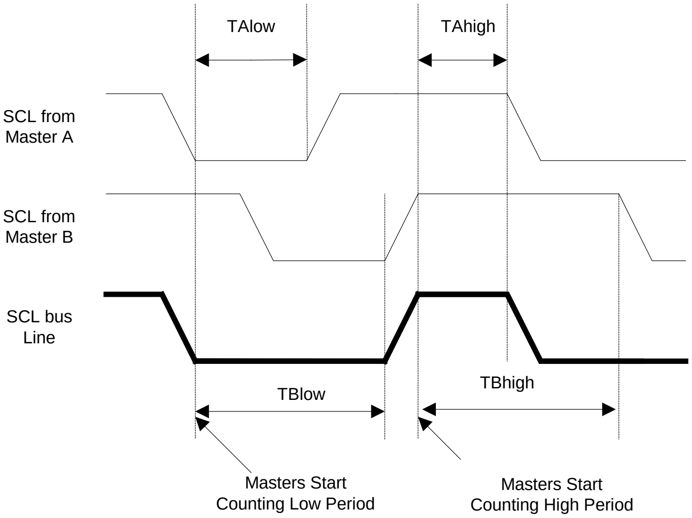


Clock Synchronization

A situation may occur that more than one master is trying to place clock signals on the bus at the same time. The resulting bus will be the wired AND of all the clock signals provided by the masters.

It is important for the bus integrity that there is a clear definition of the clock, bit by bit for all masters involved during an arbitration process.

A high to low transition on the SCL line should cause all devices involved to start counting off their low period. As soon as a device finishes counting its low period it will release the SCL line. Neverthless, the actual signal on the SCL may not transition to the high state if another master will longer low period keeps the SCL line low. In this situation the master that released the SCL line will enter the SCL high wait period. When all devices have counted off their low period, the SCL line will be released and go high. All devices concerned at this point will start counting their high periods. The first device that completes its high period count will pull the SCL line low and the cycle will start again.



Data Arbitration

A master may start a transfer only if the bus is free. Two or more devices may generate a START condition within the minimum hold time ($t_{HOLD:STA}$), resulting in a defined START condition on the bus.

Since the devices that generated the START condition may not be aware whether other masters are contending for the bus. Arbitration takes place on the SDA line while the SCL is high. When the other master is transmitting a low level on the SDA line, the master which transmits a high level will lose the arbitration and must give up bus.

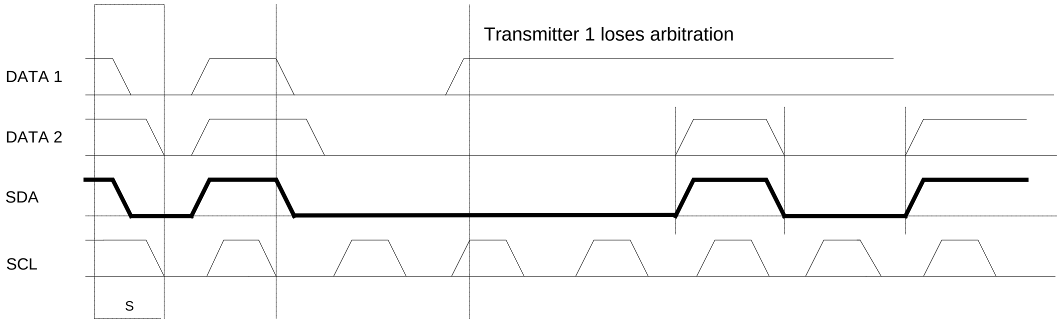
The master that lost the arbitration may continue to provide clock pulses until the current transmission bytes are finished. Arbitration in the case of two masters trying to access the same device may continue past the address byte. In this case arbitration will continue with the remaining transfer data. This mechanism requires that all TWI devices are monitoring the actual state of the SDA line during every bus transmission.

If a master also incorporates a slave mode and lose the arbitration during the address stage, it should check the actual address placed on the bus in order to determine whether another master is trying to access it. In this case the master that lost the arbitration must switch immediately to its slave mode in order to receive the rest of the message.

During each bus transmission, masters are still required to be able to recognized a repeated START condition on the bus. When detecting a repeated START condition which is not generated by itself, the device should quit the current transfer.

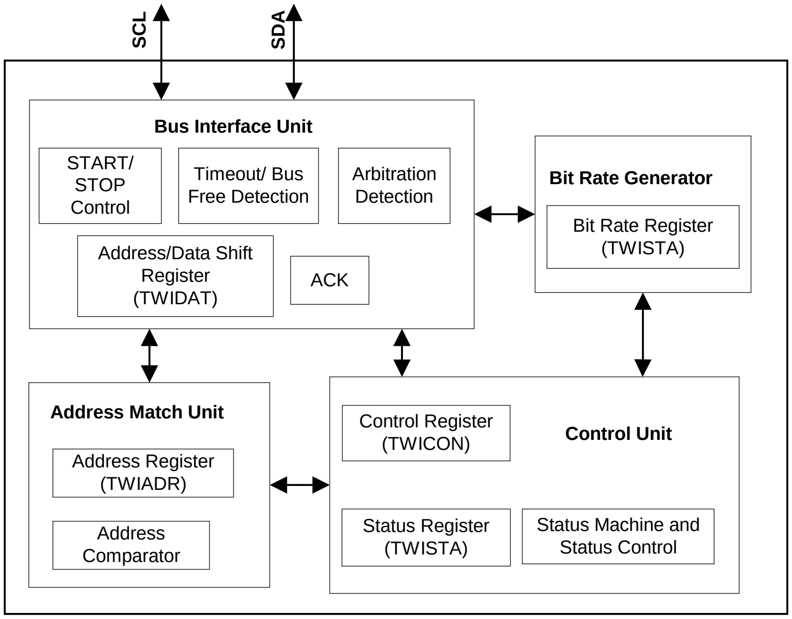
Arbitration should not occur in the following situation:

(1) A repeated START condition and data bit
(2) A STOP condition and data bit
(3) A repeated START condition and a STOP condition



8.6.3 Function Description

Detailed structure of TWI communication module is shown in the following diagram.


SCL
SDA
Bus Interface Unit
START/ STOP Control
Timeout/ Bus Free Detection
Arbitration Detection
Address/Data Shift Register (TWIDAT)
ACK
Bit Rate Generator
Bit Rate Register (TWISTA)
Address Match Unit
Address Register (TWIADR)
Address Comparator
Control Register (TWICON)
Control Unit
Status Register (TWISTA)
Status Machine and Status Control

Bus Interface Unit

This unit contains the Data and Address Shift Register (TWIDAT), a START/STOP Controller, Arbitration and Timeout detection hardware.

The TWIDAT register contains the address or data bytes to be transmitted, or the address or data bytes to be received.

The START/STOP Controller is responsible for generation and detection of START, repeated START and STOP conditions.

If SH79F3218 has initiated a transmission as Master, the Arbitration Detection hardware continuously monitors the transmission trying to determine if arbitration is in process. If TWI has lost arbitration, the Control Unit is informed Correct action can be taken and appropriate status codes generated.

SH79F3218 must hold data stable before SCL from low to high, when SH79F3218 transmit data/address.

When SH79F3218 transmit ACK/NACK, SH79F3218 will generate TWINT interrupt after SCL from low to high, and when SCL from high to low, pulling SCL low, releasing SCL when clearing TWINT interrupt.

When SH79F3218 transmit ACK/NACK, if TWINT has been cleared and SCL is still high, SDA generates jumping, TWINT interrupt will regenerate, status is 00H. The current communication of SH79F3218 will stop, the process is as the same as general 00H.

When SH79F3218 transmit ACK/NACK, if TWINT isn't cleared and SCL is still high, SDA generates jumping, then the status switch to 00H directly instead of regenerating interrupt. SH79F3218 enter into the status as slave, then the current communication will stop and can generate STA to start master transmission. Or accepting the visit of STA+ADR to own address again. SH79F3218 enter into the status as master, then the current communication will stop and can generate STA to start master transmission. Or accepting the visit of STA+ADR to own address again.

After the current communication is terminated, SH79F3218 will don't take part in the current transmission. If SH79F3218 exists as a master, please enable EFREE function to avoid entering logic deadband.

SH79F3218 defines the Bus hold high more than 50us as free mode, releasing the Bus. The function is only used in the transmission process of one packet (8 + 1 bit).

When SH79F3218 is in slave transfer mode and the first byte of transferred message is low, the function can be used. STA and RSTA is not situable for this function. If SH79F3218 generates interrupt, TFREE bit in TWICON regiser will be set (if control bit EFREE bit has been set).

When SCL line is pulled to low by slave, the communication will be stop temporarily; The master also can't pull SCL line to high. For solving this problem, TWI defines all devices that take part in transmission pull SCL line to low more than 25ms as Timeout. TOUT bit in TWICON register will be set (if control bit ETOT has been set).

TWI module will reset within 10ms and release the Bus.

Bit Rate Generator Unit

In Master mode, the baud rate si chosen from one of the four clock rates (4KHz, 16KHz, 64KHz, 100KHz (4MHz clock source), setting by CR[1:0] in TWICON register.

Address Match Unit

The Address Match unit checks if received address bytes match the 7bit address in the address register TWIADR. If the TWI General Call Recognition Enable bit is set, it will check whether match the general address 00H. Upon matching the address, the control unit will generate a appropriate action and corresponding status code.

Control Unit

The Control Unit monitors the TWI bus and generates corresponding response according to the setting of TWICON register. When an event which requires the attention of the application occurs on the TWI bus, the interrupt flag of TWI will be set, indicating that the status code of the current event will be written to TWISTA register. The status Register TWISTA only can show the communication status information when generating TWI communication interrupt; In other situations, a status code which is used to represent invalid status code in status register. Before clearing the interrupt, SCL line will hold low level. This allows the application software to complete its tasks before allowing the TWI transmission to continue.

8.6.4 Transmission Mode

TWI is a byte-oriented and interrupt based communication bus. Interrupts will be generated by all bus events, like reception of a byte or transmission of a START condition. So the application software can do other oparations during a byte transfer. Note that TWI enable (ENTWI) bit in TWICON, all interrupts enable (EA) bit in IEN0 and ETWI bit will decide together whether generating an interrupt when TWINT bit is set. If ETWI bit or EA bit is not set, the application software must poll the TWINT flag to know whether TWI event occurs.

Setting the TWINT bit indicates that a TWI transfer has completed, and it is waiting for the response of the application software. At this moment, the sattus register TWISTA contains the current status. The application software can decide which communication will be transmitted by TWI by TWICON register and TWISTA register.

The following section will introduce the four major modes of TWI communication and describe all possible status codes. These figures contain the following abbreviations:

S	: START condition
Rs	: REPEATED START condition
R	: Read bit
W	: Write bit
A	: Acknowledge bit
Ā	: Not acknowledge bit
DATA	: 8-bit data byte
P	: STOP condition
SLA	: Slave Address

The circles are used to indicate that the interrupt flag is set. The numbers in the circles show the status code held in theTWISTA, with the least 3 bit masked to zero. Before clearing TWINT, the TWI transfer will be suspended, the application must decide whether to continue or stop the current transfer. For each status code, the required software action and details of the following serial transfer are given.

Master Transmitter Mode

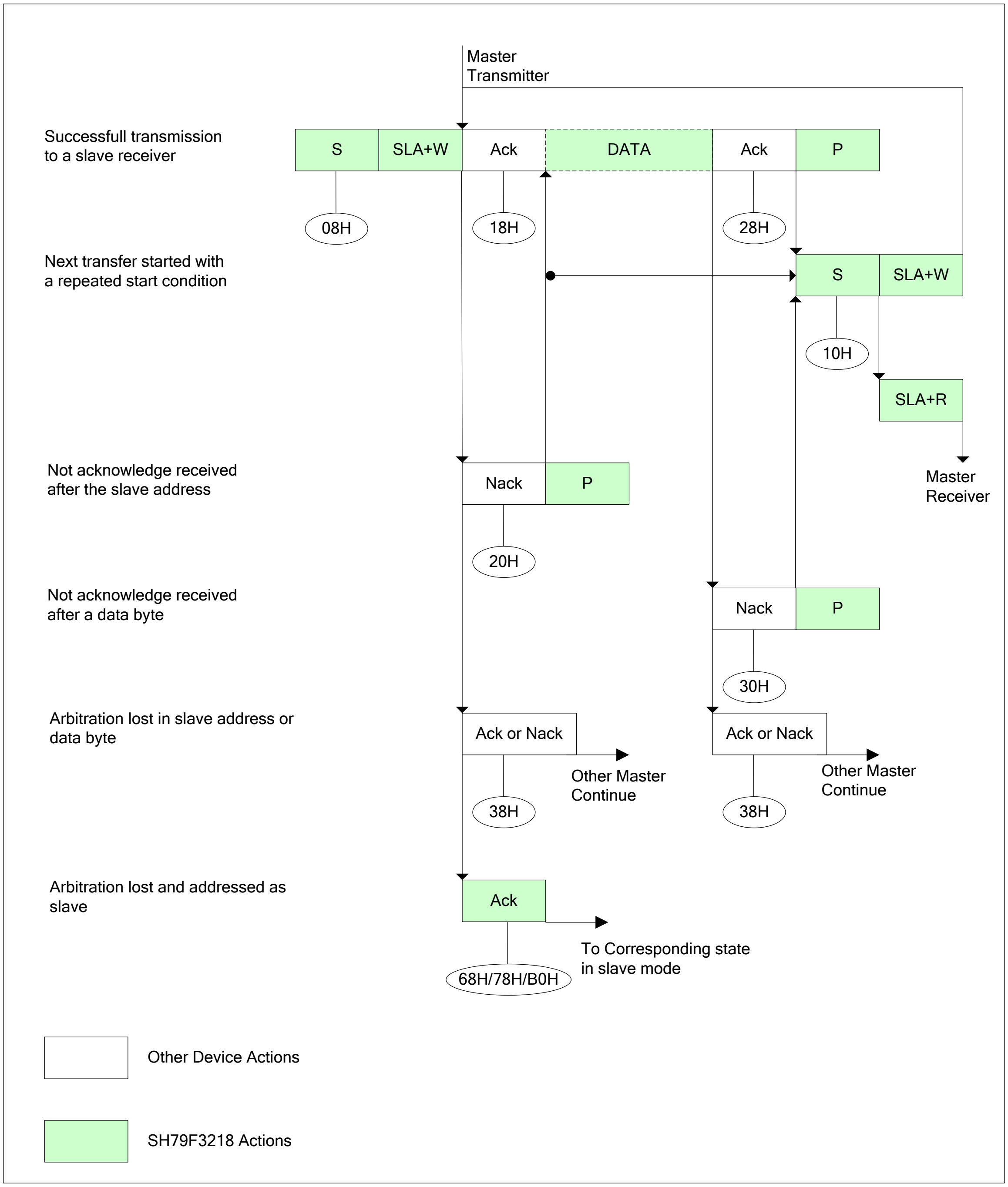
In the Master Transmitter mode, a number of data bytes are transmitted to a slave receive.In order to enter a Master mode,a START condition must be transmitted, a following SLA + W address packet determines MT has entered.

By setting ENTWI and STA in TWICON register, clearing STO and TWINT, the TWI logic will test TWI bus and generate a start condition as soon as the bus becomes free. When a START condition is transmitted, the interrupt (TWINT) will be set, the status register TWISTA is 08H. The interrupt service routine should load TWIDAT with the slave address and the data direction bit (SLA + W). Clearing TWINT flag before start the next transfer.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, TWINT will be set, and a number of status codes in TWISTA are possible. There are 18H, 20H and 38H for the master mode, and also 68H, 78H and B0H for the slave mode.

Status Code for Master Transmitter Mode

Status Code	Status of TWI bus and Hardware Interface	Application Software Response					Next Action Taken by Hardware
		To/From TWIDAT	Control Bit Operation				
			STA	STO	TWINT	AA	
08H	A START Condition has been transmitted	Load SLA + W	X	0	0	X	Transmit SLA+W, receive ACK
10H	A repeated START Condition has been transmitted	Load SLA + W	X	0	0	X	Transmit SLA+W, receive ACK
		Load SLA + R	X	0	0	X	Transmit SLA+R, TWI will switch to Master Receiver mode
18H	SLA + W has been transmitted; ACK has been received	Load data byte	0	0	0	X	Transmit data, receive ACK
		No TWIDAT action	1	0	0	X	Transmit the repeated Start condition
			0	1	0	X	Transmit STOP condition; Clearing STO flag
			1	1	0	X	Transmit STOP condition, and then transmit START condition; clearing STO
20H	SLA + W has been transmitted; NACK has been received	Load data byte	0	0	0	X	Transmit data, receive ACK
		No TWIDAT action	1	0	0	X	Transmit the repeated Start condition
			0	1	0	X	Transmit STOP condition; Clearing STO flag
			1	1	0	X	Transmit STOP condition, and then transmit START condition; clearing STO
28H	Data byte in TWIDAT has been transmitted; ACK has been received	Load data byte	0	0	0	X	Transmit data, receive ACK
		No TWIDAT action	1	0	0	X	Transmit the repeated Start condition
			0	1	0	X	Transmit STOP condition; Clearing STO flag
			1	1	0	X	Transmit STOP condition, and then transmit START condition; clearing STO
30H	Data byte in TWIDAT has been transmitted; ACK has been received	Load data byte	0	0	0	X	Transmit data, receive ACK
		No TWIDAT action	1	0	0	X	Transmit the repeated Start condition
			0	1	0	X	Transmit STOP condition; Clearing STO flag
			1	1	0	X	Transmit STOP condition, and then transmit START condition; clearing STO
38H	Lose arbitration in SLA + W or data transmit	No TWIDAT action	0	0	0	X	Releasing TWI bus; Entering not addressed slave mode
			1	0	0	X	Transmit START condition when bus is free


Master
Transmitter
Successfull transmission
to a slave receiver
S
SLA+W
Ack
DATA
Ack
P
08H
18H
28H
Next transfer started with
a repeated start condition
S
SLA+W
10H
SLA+R
Master
Receiver
Not acknowledge received
after the slave address
Nack
P
20H
Not acknowledge received
after a data byte
Nack
P
30H
Arbitration lost in slave address or
data byte
Ack or Nack
Other Master
Continue
38H
Ack or Nack
Other Master
Continue
38H
Arbitration lost and addressed as
slave
Ack
To Corresponding state
in slave mode
68H/78H/B0H
Other Device Actions
SH79F3218 Actions

Master Receiver Mode

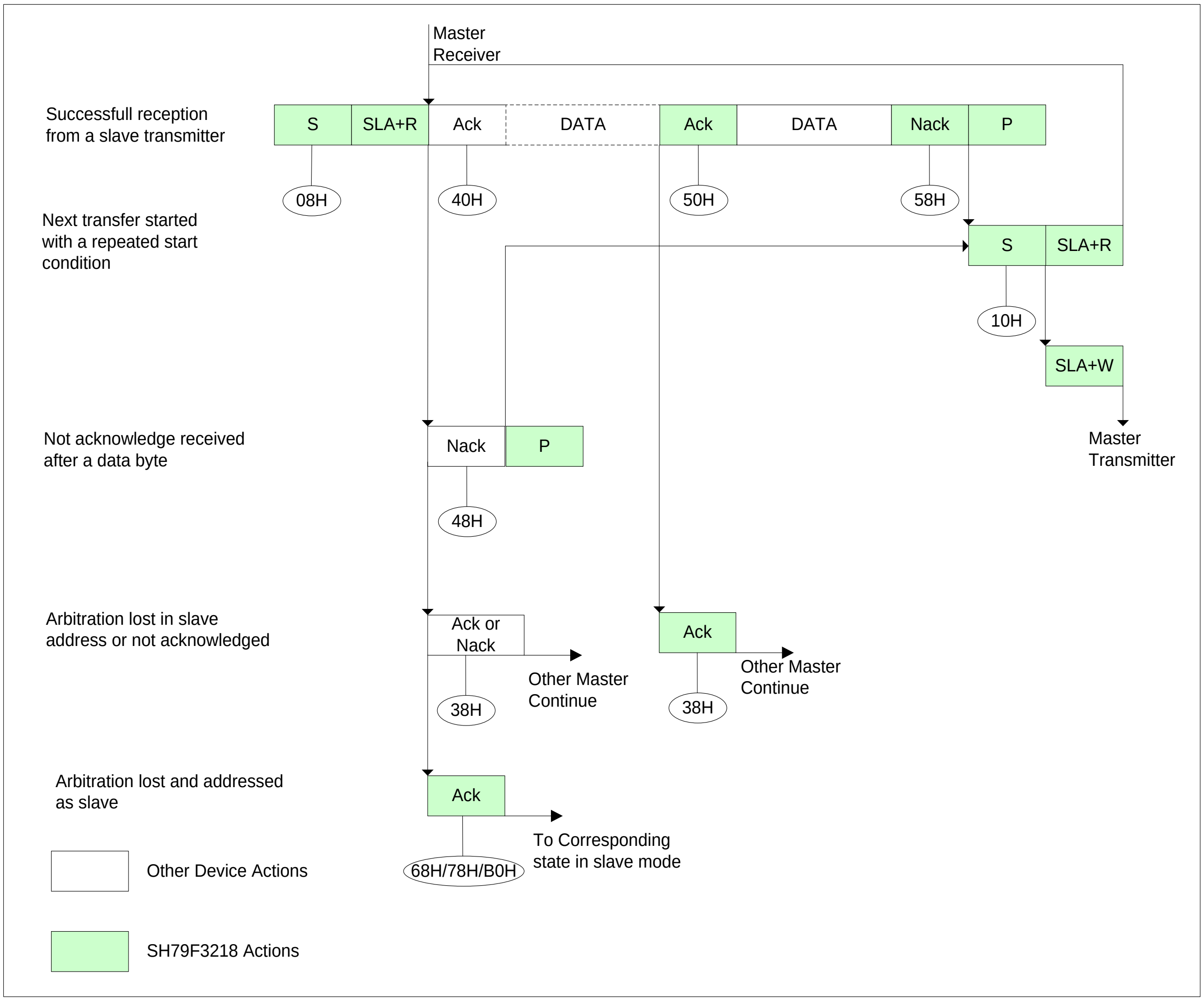
In the Master Receiver mode, a number of data bytes are received from a slave. In order to enter a Master mode, a STARTcondition must be transmitted, a following SLA + R address packet determines MR has entered.

Clearing STO and TWINT by setting ENTWI and STA in TWICON register. TWI logic will test TWI bus and generate a START condition as soon as the bus becomes free. When a START condition is transmitted, the interrupt (TWINT) will be set, the status register TWISTA is 08H. The interrupt service routine should load TWIDAT with the slave address and the data direction bit (SLA + R). Clearing TWINT flag before start the next transfer.

When the slave address and the direction bit have been transmitted and an acknowledgment bit has been received, TWINT will be set, and a number of status codes in TWISTA are possible. There are 40H, 48H and 38H for the master mode, and also 68H, 78H and B0H for the slave mode.

Status Code for Master Receiver Mode

Status Code	Status of TWI bus and Hardware Interface	Application Software Response					Next Action Taken by Hardware
		To/From TWIDAT	Control Bit Operation				
			STA	STO	TWINT	AA	
08H	A START Condition has been transmitted	Load SLA + R	X	0	0	X	Transmit SLA+R, receive ACK
10H	A repeated START Condition has been transmitted	Load SLA + R	X	0	0	X	Transmit SLA+R, receive ACK
		Load SLA + W	X	0	0	X	Transmit SLA + W, TWI will switch to Master transmitter mode
38H	Lose arbitration when transmitting SLA + R or NACK	No TWIDAT action	0	0	0	X	Releasing TWI bus; Entering not address slave mode
			1	0	0	X	Transmit START condition when bus is free
40H	SLA + R has been transmitted; ACK has been received	No TWIDAT action	0	0	0	0	Receive data, go back to NACK
			0	0	0	1	Receive data, go back to ACK
48H	SLA + R has been transmitted; NACK has been received	No TWIDAT action	1	0	0	X	Transmit the repeated Start condition
			0	1	0	X	Transmit STOP condition; Clearing STO flag
			1	1	0	X	Transmit STOP condition, and then transmit START condition; clearing STO
50H	Data byte has been received; ACK has responded	Read data bit	0	0	0	0	Receive data, go back to NACK
			0	0	0	1	Receive data, go back to ACK
58H	Data byte has been received; NACK has responded	Read data bit	1	0	0	X	Transmit the repeated Start condition
			0	1	0	X	Transmit STOP condition; Clearing STO flag
			1	1	0	X	Transmit STOP condition, and then transmit START condition; clearing STO



Slave Transmitter Mode

In the Slave Transmitter mode, a number of data bytes are transmitted to a master receive.In order to initialize Slave transmitter mode, TWICON register and TWIADR register must be initialized: set ENTWI bit and AA bit in TWICON register, clearing STA, STO and TWINT; The high 7-bit in TWIADR register is used to prepare the corresponding address for SH79F3218. If GC is set, SH79F3218 will respond the general address (00H); Otherwise, SH79F3218 will not respond the address.

When TWIADR and TWICON are initialized, SH79F3218 will be waiting for the response to itself address or general address (if GC is set).

If the direction bit is “R”, then TWI enter to the Slave transmitter mode. Otherwise, TWI will enter to the Slave receiver mode. After its own slave address and read bit have been received, TWINT will be set and a valid status code can be read from TWISTA.

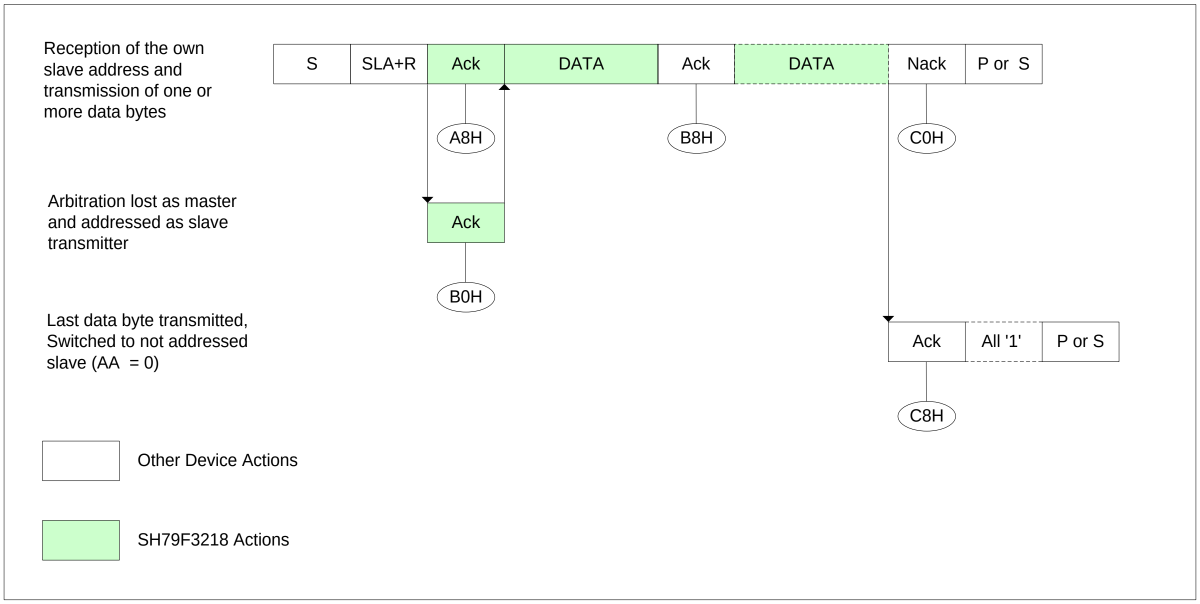
If the AA bit is cleared during a transfer, TWI will transmit the last byte. State C0H or C8H will be entered, depending on whether the master receiver transmits a NACK or ACK after the final byte. TWI bus will switch to the not address slave mode, and will ignore the master if it continues the transfer. Thus the master receiver will receive all “1” as serial data. State C8H is entered if the master demands additional data bytes (by transmitting ACK), even though the slave has transmitted the last byte.

Status Code for Slave Transmitter Mode

Status Code	Status of TWI bus and Hardware Interface	Application Software Response					Next Action Taken by Hardware
		To/From TWIDAT	Control Bit Operation				
			STA	STO	TWINT	AA	
A8H	Own SLA + R has been received; ACK has been received	Load data byte	X	0	0	0	Transmit the last data byte and ACK will be received
			X	0	0	1	Transmit data byte and ACK will be received

B0H	Arbitration lost in SLA + R/W as master;
	Own SLA + R has been received;
	ACK has been received
		Load data byte	X	0	0	0	Transmit the last data byte and ACK will be received

			X	0	0	1	Transmit data byte and ACK will be received
B8H	TWIDAT data has transmitted; ACK has been received	Load data byte	X	0	0	0	Transmit the last data byte and ACK will be received
			X	0	0	1	Transmit data byte and ACK will be received
C0H	TWIDAT data has transmitted; NACK has been received	No TWIDAT action	0	0	0	0	Switched to not addressed SLA mode; No recognition of own SLA or General address
			0	0	0	1	Switched to not addressed SLA mode; Own SLA will be recognized, general address will be recognized if TWIADR.0 = 1
			1	0	0	0	Switched to not addressed SLA mode; No recognition of own SLA or General address; Transmit “START condition” when bus is free
			1	0	0	1	Switched to not addressed SLA mode; Own SLA will be recognized, general address will be recognized if TWIADR.0 = 1; Transmit “START condition” when bus is free
C8H	Last data byte in TWIDAT has been transmitted (AA = 0) ; ACK has been received	No TWIDAT action	0	0	0	0	Switched to not addressed SLA mode; No recognition of own SLA or General address
			0	0	0	1	Switched to not addressed SLA mode; Own SLA will be recognized, general address will be recognized if TWIADR.0 = 1
			1	0	0	0	Switched to not addressed SLA mode; No recognition of own SLA or General address; Transmit “START condition” when bus is free
			1	0	0	1	Switched to not addressed SLA mode Own SLA will be recognized, general address will be recognized if TWIADR.0 = 1; Transmit “START condition” when bus is free


Reception of the own slave address and transmission of one or more data bytes
S
SLA+R
Ack
DATA
Ack
DATA
Nack
P or S
A8H
B8H
C0H
Arbitration lost as master and addressed as slave transmitter
Ack
B0H
Last data byte transmitted, Switched to not addressed slave (AA = 0)
Ack
All '1'
P or S
C8H
Other Device Actions
SH79F3218 Actions

Slave Receiver Mode

In the Slave receiver mode, a number of data bytes are received from a master transmitter. In order to initialize Slave receiver mode, TWICON register and TWIADR register must be initialized: set ENTWI bit and STA bit in TWICON register, clearing STO and TWINT; The high 7-bit in TWIADR register is used to prepare the corresponding address for SH79F3218. If GC is set, SH79F3218 will respond the general address (00H); Otherwise, SH79F3218 will not respond the address.

When TWIADR and TWICON are initialized, SH79F3218 will be waiting for the response to itself address or general address (if GC is set).

If the direction bit is “W”, then TWI enter to the Slave receiver mode. Otherwise, TWI will enter to the Slave transmitter mode. After its own slave address and write bit have been received, TWINT will be set and a valid status code can be read from TWISTA.

If the AA bit is cleared during a transfer, TWI will receive the last byte and respond NACK information. Responding NACK indicates the current slave can’t receive more bytes. When AA = 0, SH79F3218 can’t respond the visit to its own address; However, SH79F3218 still monitors the bus status, and address recognition may resume at any time by setting AA. This implies that the AA bit may be used to temporarity isolate SH79F3218 from the bus.

When SH79F3218 is in Slave Receiver mode, the minimum receive frequency is 4.5KHz, less than 4.5KHz, unable to properly receive data.

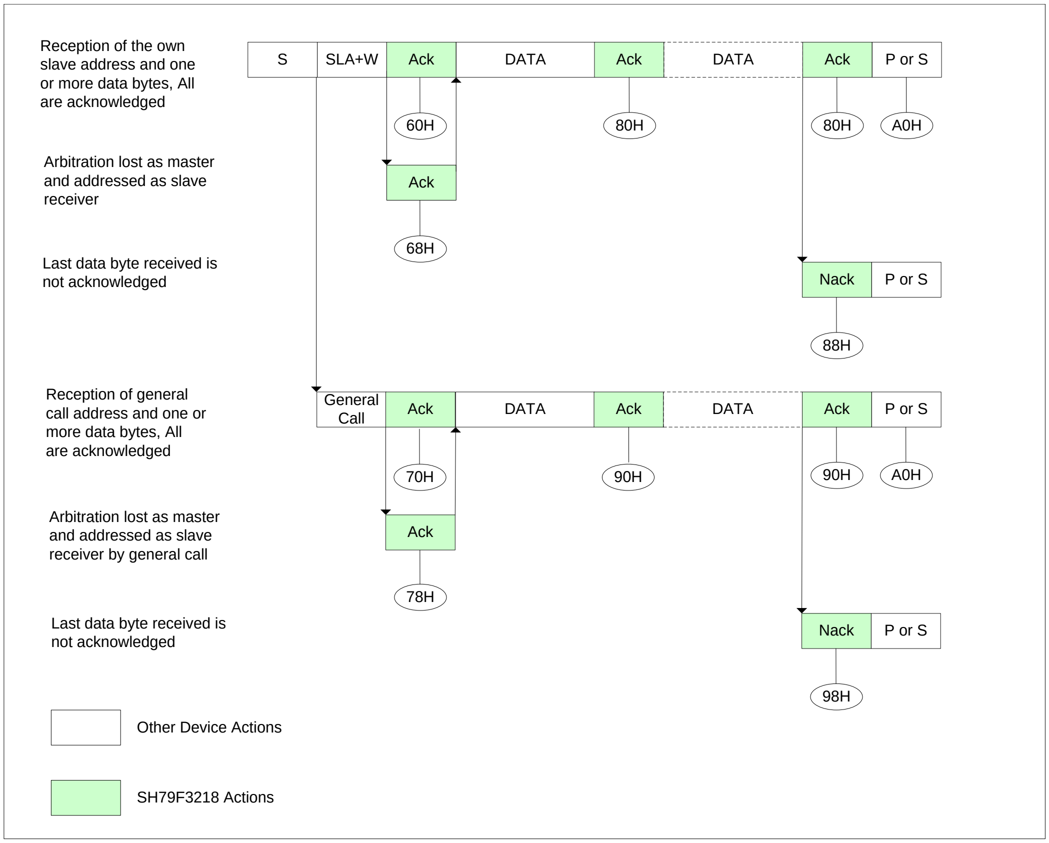
Status Code for Slave Receiver Mode

Status Code	Status of TWI bus and Hardware Interface	Application Software Response					Next Action Taken by Hardware
		To/From TWIDAT	Control Bit Operation				
			STA	STO	TWINT	AA	
60H	Own SLA + W has been received; ACK has been received	No TWIDAT action	X	0	0	0	Receive data byte; Transmit NACK
			X	0	0	1	Receive data byte; Transmit ACK
68H	Arbitration lost in SLA + R/W as master; Own SLA + W has been received； ACK has been received	No TWIDAT action	X	0	0	0	Receive data byte; Transmit NACK
			X	0	0	1	Receive data byte; Transmit ACK
70H	General address has been received; ACK has been received	No TWIDAT action	X	0	0	0	Receive data byte; Transmit NACK
			X	0	0	1	Receive data byte; Transmit ACK

(to be continued)

(continue)

78H	Arbitration lost in SLA + R/W as master; General address has been received; ACK has been received	No TWIDAT action	X	0	0	0	Receive data byte; Transmit NACK
			X	0	0	1	Receive data byte; Transmit ACK
80H	Previously addressed with own SLA address; Data has been received; ACK has been received	Read data byte	X	0	0	0	Receive data byte; Transmit NACK
			X	0	0	1	Receive data byte; Transmit ACK
88H	Previously addressed with own SLA address; Data has been received; NACK has been received	Read data byte	0	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or general address
			0	0	0	1	Switched to not addressed SLA mode; SLA will be recognized, general address will be recognized if TWIADR.0 = 1
			1	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or general address; Transmit “START condition” when bus is free
			1	0	0	1	Switched to not addressed SLA mode; SLA will be recognized, general address will be recognized if TWIADR.0 = 1; Transmit “START condition” when bus is free
90H	Previously addressed with General address; Data has been received; ACK has been received	Read data byte	X	0	0	0	Receive data byte; Transmit NACK
			X	0	0	1	Receive data byte; Transmit ACK
98H	Previously addressed with General address; Data has been received; NACK has been received	Read data byte	0	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or general address
			0	0	0	1	Switched to not addressed SLA mode; SLA will be recognized, general address will be recognized if TWIADR.0 = 1
			1	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or general address; Transmit “START condition” when bus is free
			1	0	0	1	Switched to not addressed SLA mode; SLA will be recognized, general address will be recognized if TWIADR.0 = 1; Transmit “START condition” when bus is free
A0H	A STOP condition or repeated START condition has been received while still addressed as slave receiver	No TWIDAT action	0	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or general address
			0	0	0	1	Switched to not addressed SLA mode; SLA will be recognized, general address will be recognized if TWIADR.0 = 1
			1	0	0	0	Switched to not addressed SLA mode; no recognition of own SLA or general address; Transmit “START condition” when bus is free
			1	0	0	1	Switched to not addressed SLA mode; SLA will be recognized, general address will be recognized if TWIADR.0 = 1; Transmit “START condition” when bus is free


Reception of the own slave address and one or more data bytes, All are acknowledged
S
SLA+W
Ack
DATA
Ack
DATA
Ack
P or S
60H
80H
80H
A0H
Arbitration lost as master and addressed as slave receiver
Ack
68H
Last data byte received is not acknowledged
Nack
P or S
88H
Reception of general call address and one or more data bytes, All are acknowledged
General Call
Ack
DATA
Ack
DATA
Ack
P or S
70H
90H
90H
A0H
Arbitration lost as master and addressed as slave receiver by general call
Ack
78H
Last data byte received is not acknowledged
Nack
P or S
98H
Other Device Actions
SH79F3218 Actions

Other Modes

Except for the above status codes, there are two status codes without specific TWI status. The status 0F8H indicates that no relevant information is available because TWINT is not set. This occurs between other states, and when the TWINT is not involved in a serial transfer.

Status 0x00 indicates that a bus error has occurred during a TWI bus serial transfer. A bus error is caused when a START or STOP condition occurs at an illegal position in the format frame. Examples of such illegal positions are during the serial transfer of an address byte, a data byte, or an acknowledge bit. A bus error may also be caused when external interference disturbs the internal TWI bus signals. When a bus error occurs, TWINT will be set. To recover from a bus error, the STO flag must be set and TWINT must be cleared. This will cause SH79F3218 to enter the not addressed slave mode and to clear the STO flag. SDA and SCL lines will be released, and no STOP condition is transmitted.

Status Code of Other Modes

Status Code	Status of TWI bus and Hardware Interface	Application Software Response					Next Action Taken by Hardware
		To/From TWIDAT	Control Bit Operation				
			STA	STO	TWINT	AA	
F8H	Without valid status code; TWINT = 0	No TWIDAT action	No TWICON action				Wait or proceed current transfer
00H	Bus error during Master or selected slave modes, due to an illegal START or STOP condition. Interface cause TWI internal logic mess.	No TWIDAT action	0	1	0	X	Only the internal hardware is affected in the Master or addressed Slave modes. In all cases, the bus is released and TWI bus is switched to the not addressed Slave mode, STO is reset

8.6.5 Register

Table 8.31 TWI Control Register

F8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TWICON	TOUT	ENTWI	STA	STO	TWINT	AA	TFREE	EFREE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**TOUT**	**Bus line timeout flag** 0: No timeout occurred 1: Set by hardware when TWI bus low level exceeds the timeout period (25ms). It must be cleared by software.
6	**ENTWI**	**TWI Enable bit** 0: Disable TWI 1: Enable TWI
5	**STA**	**START condition flag** 0: No START condition is transmitted 1: Transmit START condition when the bus is free
4	**STO**	**STOP condition flag** 0: No STOP condition is transmitted 1: Transmit STOP condition as Master; Don’t transmit STOP condition as Slave, but the status will recover to not addressed slave status. Hardware will clear this flag automatically.
3	**TWINT**	**TWI serial interrupt flag** 0: No TWI serial interrupt occurred 1: Set by hardware when generate other states except for 0F8H in TWI communication and must be cleared by software
2	**AA**	**Statement Acknowledge Flag** 0: Return NACK signal (high level on SDA) 1: Return ACK signal (low level on SDA)
1	**TFREE**	**SCL High Level Timeout Flag** 0: No high level timeout occurred 1: Set by hardware when SCL high level exceeds the timeout period (50us), it must be cleared by sofaware.
0	**EFREE**	**SCL High Level Timeout Enable bit** 0: Disable SCL high level timeout detection 1: Enable SCL high level timeout detection

Note: *TOUT, TWINT, TFREE share one interrupt vector, all of them can trigger TWI interrupt.*

Table 8.32 TWI Status Register

F9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TWISTA	TWISTA.7	TWISTA.6	TWISTA.5	TWISTA.4	TWISTA.3	CR.1	CR.0	ETOT
R/W	R	R	R	R	R	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIIN)	1	1	1	1	1	0	0	0

Bit Number	Bit Mnemonic	Description
7-3	**TWISTA[7:3]**	**TWI Status bit of Serial Communication** See operation mode for details
2-1	**CR[0:1]**	**TWI Serial bit rate** 00: f_{OSC}/6/1024 01: f_{OSC}/6/256 10: f_{OSC}/6/64 11: f_{OSC}/6/42 When SH79F3218 is in Master mode, the hold time of STA, STO and repeated STA is related to the transfer frequency which is selected by CR[1:0]
0	**ETOT**	**Timeout Enable Bit** 0: Disable Timeout detection 1: Enable Timeout detection

Table 8.33 TWI Address Register

FAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TWIADR	TWA.6	TWA.5	TWA.4	TWA.3	TWA.2	TWA.1	TWA.0	GC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-1	**TWA[6:0]**	**TWI Address Configuration bit** Configure SH79F3218 as the address in Slave mode
0	**GC**	**General Address Enable bit** 0: Disable general address 1: Enable general address

Table 8.34 TWI Data Register

FDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TWIDAT	TWIDAT.7	TWIDAT.6	TWIDAT.5	TWIDAT.4	TWIDAT.3	TWIDAT.2	TWIDAT.1	TWIDAT.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**TWIDAT[7:0]**	TWI Communication Data Register

Table 8.35 System Clock Control Register

B2H，Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	32k_SPDUP	CLKS1	CLKS0	SCMIF	HFON	FS	-	-
R/W	R/W	R/W	R/W	R	R/W	R/W	-	-
Reset Value (POR/WDT/LVR/PIN)	1	1	1	0	0	0	-	-

Bit Number	Bit Mnemonic	Description
7	**32k_SPDUP**	**32.768kHz oscillator speed up mode control bit** 0: 32.768kHz oscillator normal mode, cleared by software. 1: 32.768kHz oscillator speed up mode, set by hardware or software. This control bit is set by hardware automatically in all kinds of RESET such as Power on reset, watch dog reset etc. to speed up the 32.768kHz Oscillator oscillating, shorten the 32.768kHz oscillator start-oscillating time. And this bit also can be set or cleared by software if necessary. Such as set before entering Power-down mode and cleared when Power-down mode wakes up. It should be noticed that turning off 32.768kHz oscillator speed up (clear this bit) could reduce the system power consumption. Only when code option OP_OSC is 1010, this bit is valid. (32.768kHz oscillator is selected, Refer to **code option** section for details)
6-5	**CLKS[1:0]**	**SYSCLK Prescaler Register** 00: $f_{SYS} = f_{OSCS}$ 01: $f_{SYS} = f_{OSCS}/2$ 10: $f_{SYS} = f_{OSCS}/4$ 11: $f_{SYS} = f_{OSCS}/12$ If 32.768kHz oscillator is selected as OSCSCLK, these control bits is invalid.
3	**HFON**	**OSC1CLK On-Off control Register** 0: turn off OSC1CLK 1: turn on OSC1CLK Only when code option OP_OSC is 0011, 0110, 1010 this bit is valid. (32.768kHz oscillator/128kHz internal RC is selected, refer to **code option** section for details)。
2	**FS**	**Frequency Select Register** 0: 32.768kHz/128kHz is selected as OSCSCLK. 1: OSC1CLK is selected as OSCSCLK. Only when code option OP_OSC is 0011, 0110, 1010 this bit is valid. (32.768kHz oscillator/128kHz internal RC is selected, refer to **code option** section)

8.7 Buzzer

8.7.1 Feature

- Output a signal (square wave) used for tones such as confirmation tone
- Output a adjustable cycle square wave signal

8.7.2 Register

Table 8.36 Buzzer Output Control Register

BDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZCON	BUZEN	BUSLE	-	-	BCA1	BCA0	-	-
R/W	R/W	R/W	-	-	R/W	R/W		
Reset Value (POR/WDT/LVR/PIN)	0	0	-	-	0	0		

Bit Number	Bit Mnemonic	Description
7	**BUZEN**	**Buzzer control bit** 0: Disable Buzzer 1: Enable Buzzer
6	**BUSLE**	**Enable buzzer output control bit** 0: BUZ1 is Buzzer output (BUZ2 is I/O) 1: BUZ2 is Buzzer output (BUZ1 is I/O)
3-2	**BCA[2:0]**	**Buzzer output carrier frequency control bits** 00: system clock/256 01: system clock/64 10: system clock/16 11:system clock/4

Table 8.37 Buzzer Period control register (BUZP)

AFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZP	BZP.7	BZP.6	BZP.5	BZP.4	BZP.3	BZP.2	BZP.1	BZP.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**BZP[7:0]**	**Buzzer Period 8 bit register**

Buzzer Calculation formula of the output square wave frequency:

Buzz Frequency = Fsys/BCA[2:0]/BUZP[7:0]

8.8 Low Power Detect (LPD)

8.8.1 Feature

- Low power detect and generate interrupt
- LPD detect voltage is selectable

The low power detect (LPD) is used to monitor the supply voltage and generate an internal flag if the voltage decrease below the specified value. It is used to inform CPU whether the power is shut off or the battery is used out, so the software may do some protection action before the voltage drop down to the minimal operation voltage.

8.8.2 Register

Table 8.38 Low Power Detection Control Register

B3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPDCON	LPDEN	LPDF*	-	-	-	LPDS2	LPDS1	LPDS0
R/W	R/W	R/W	-	-	-	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	-	-	-	0	0	0

**: LPDF Software can only clear 0, can not set 1.*

Bit Number	Bit Mnemonic	Description
7	**LPDEN**	**LPD Enable bit** 0: Disable lower power detection 1: Enable lower power detection
6	**LPDF**	**LPD status Flag bit** 0: No LPD happened, clear by hardware or software. (the current voltage is higher than the voltage of LPD set in LPDS[1:0]) 1: LPD happened, set by hardware. (the current voltage is lower than the voltage of LPD set in LPDS[1:0])
2-0	**LPDS[2:0]**	**LPD Voltage Select bit** 000: 2.8V 001: 3.0V 010: 3.2V 011: 3.4V 100: 3.6V 101: 3.8V 110: 4.0V 111: 4.2V

8.9 Low Voltage Reset (LVR)

8.9.1 Feature

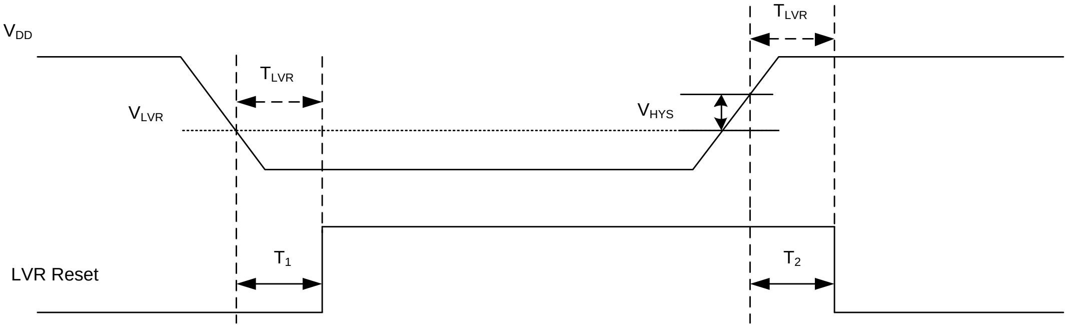
- Enabled by the code option and V_{LVR} is 3.7V, 3.1V or 2.8V
- LVR de-bounce timer T_{LVR} is 30-60μs
- When the power supply voltage is lower than the set voltage V_{LVR}, it will cause the internal reset

The LVR function is used to monitor the supply voltage and generate an internal reset in the device when the supply voltage below the specified value V_{LVR}. The LVR de-bounce timer T_{LVR} is about 30μs-60μs.

The LVR circuit has the following feature when the LVR function is enabled: (T_1 means the time of the supply voltage lower V_{LVR}, T_2 means the time of the supply voltage higher V_{LVR}+V_{HYS})

Generates a system reset when $V_{DD} \leq V_{LVR}$ and $T_1 \geq T_{LVR}$;

Cancels the system reset when $V_{DD} > V_{LVR}$ or $V_{DD} < V_{LVR}$, but $T_1 < T_{LVR}$. V_{HYS} = 0.09V-0.11V.



V_{DD} is the power supply voltage, V_{LVR} is LVR detection voltage, V_{HYS} is low voltage reset delay voltage

The LVR function is enabled by the code option.

It is typically used in AC or large capacity battery applications, where heavy loads may be switched on and cause the MCU supply-voltage temporarily falls below the minimum specified operating voltage.

Low voltage reset can be applied to this, protecting system generates valid reset in the below set voltages

8.10 Watchdog Timer (WDT) and Reset State

8.10.1 Features

- Auto detect Program Counter (PC) over range, and generate WDT Reset
- WDT runs even in the Power-Down mode
- Selectable different WDT overflow frequency

OVL Reset

To enhance the anti-noise ability, SH79F3218 built in Program Counter (PC) over range detect circuit, if program counter value is larger than flash romsize, or detect operation code equal to A5H which is not exist in 8051 instruction set, a OVL reset will be generate to reset CPU, and set WDOF bit. So, to make use of this feature, you should fill unused flash rom with A5H.

Watchdog Timer

The watchdog timer is a down counter, and its clock source is an independent built-in RC oscillator, so it always runs even in the Power-Down mode. The watchdog timer will generate a device reset when it overflows. It can be enabled or disabled permanently by the code option.

The watchdog timer control bits (WDT.2-0) are used to select different overflow frequency. The watchdog timer overflow flag (WDOF) will be automatically set to “1” by hardware when overflow happens. To prevent overflow happen, by reading or writing the WDT register RSTSTAT, the watchdog timer should re-count before the overflow happens.

There are also some reset flags in this register as below:

8.10.2 Register

Table 8.39 Reset Control Register

B1H, Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RSTSTAT	WDOF	-	PORF	LVRF	-	WDT.2	WDT.1	WDT.0
R/W	R/W	-	R/W	R/W	-	R/W	R/W	R/W
Reset Value (POR)	0	-	1	0	-	0	0	0
Reset Value (WDT)	1	-	u	u	-	0	0	0
Reset Value (LVR)	u	-	u	1	-	0	0	0
Reset Value (PIN)	u	-	u	u	-	0	0	0

Bit Number	Bit Mnemonic	Description
7	**WDOF**	**Watch Dog Timer Overflow or OVL Reset Flag** Set by hardware when WDT overflow or OVL reset happened, cleared by software or Power On Reset 0: Watch Dog not overflows or no OVL reset generated 1: Watch Dog overflow or OVL reset occurred
5	**PORF**	**Power On Reset Flag** Set only by Power On Reset, cleared only by software 0: No Power On Reset 1: Power On Reset occurred
4	**LVRF**	**Low Voltage Reset Flag** Set only by Low Voltage Reset, cleared by software or Power On Reset 0: No Low Voltage Reset occurs 1: Low Voltage Reset occurred
2-0	**WDT[2:0]**	**WDT Overflow period control bit** 000: Overflow period minimal value = 4096 ms 001: Overflow period minimal value = 1024 ms 010: Overflow period minimal value = 256 ms 011: Overflow period minimal value = 128 ms 100: Overflow period minimal value = 64ms 101: Overflow period minimal value = 16ms 110: Overflow period minimal value = 4ms 111: Overflow period minimal value = 1ms ***Notes:*** *If WDT_opt is enable in application, you must clear WatchDog periodically, and the interval must be less than the minimum value listed above.*

8.11 Power Management

8.11.1 Feature

- Two power saving modes: Idle mode and Power-Down mode
- Two ways to exit Idle and Power-Down mode: interrupt and reset

To reduce power consumption, SH79F3218 supplies two power saving modes: Idle mode and Power-Down mode. These two modes are controlled by PCON & SUSLO register.

8.11.2 Idle Mode

In this mode, the clock of CPU is frozen, the program execution is halted, and the CPU will stop at a defined state. But the peripherals continue to be clocked. When entering idle mode, all the CPU status before entering will be preserved. Such as: PSW, PC, SFR & RAM are all retained.

By two consecutive instructions: setting SUSLO register as 0x55, and immediately followed by setting the IDL bit in PCON register, will make SH79F3218 enter Idle mode. If the consecutive instruction sequence requirement is not met, the CPU will clear either SUSLO register or IDL bit in the next machine cycle. And the CPU will not enter Idle mode. The setting of IDL bit will be the last instruction that CPU executed.

There are two ways to exit Idle mode:

(1) An interrupt generated. After warm-up time, the clock of the CPU will be restored, and the hardware will clear SUSLO register and IDL bit in PCON register. Then the program will execute the interrupt service routine first, and then jumps to the instruction immediately following the instruction that activated Idle mode.

(2) Reset signal (logic low on the RESET pin, WDT RESET if enabled, LVR RESET if enabled), this will restore the clock of the CPU, the SUSLO register and the IDL bit in PCON register will be cleared by hardware, finally the SH79F3218 will be reset. And the program will execute from address 0000H. The RAM will keep unchanged and the SFR value might be changed according to different function module.

8.11.3 Power-Down Mode

The Power-Down mode places the SH79F3218 in a very low power state.

When single clock signal input (OP_OSC[3:0] is 0000 or 1110), Power-Down mode will stop all the clocks including CPU and peripherals. When double clock signa input (OP_OSC[3:0] is 0011, 0110, 1010 or 1101), if system clock is 32.768kHz or 128kHzRC, Power-Down mode will stop all the clocks including CPU and peripherals. If high frequency oscillator is used as system clock, 32.768kHz or 128kHzRC clock used in Timer3 will be opened in Power-Down mode. In Power-Down mode, if WDT is enabled, WDT block will keep on working. When entering Power-Down mode, all the CPU status before entering will be preserved. Such as: PSW, PC, SFR & RAM are all retained.

By two consecutive instructions: setting SUSLO register as 0x55, and immediately followed by setting the PD bit in PCON register, will make SH79F3218 enter Power-Down mode. If the consecutive instruction sequence requirement is not met, the CPU will clear either SUSLO register or PD bit in the next machine cycle. And the CPU will not enter Power-Down mode.

The setting of PD bit will be the last instruction that CPU executed.

***Note:** If IDL bit and PD bit are set simultaneously, the SH79F3218 enters Power-Down mode. The CPU will not go in Idle mode when exiting from Power-Down mode, and the hardware will clear both IDL & PD bit after exit form Power-Down mode.*

There are three ways to exit the Power-Down mode:

(1) An active external Interrupt (such as INT2, INT3 & INT4) and LPD interrupt will make SH79F3218 exit Power-Down mode. The oscillator will start after interrupt happens, after warm-up time, the clocks of the CPU and peripheral will be restored, the SUSLO register and PD bit in PCON register will be cleared by hardware. Program execution resumes with the interrupt service routine. After completion of the interrupt service routine, the instructions which jumped to enter Power-Down mode will continue to run.

(2) Timer3 interrupt will make SH79F3218 exit Power-Down mode when 32.768kHz or 128kHz RC is the clock source. The oscillator will start after interrupt happens, after warm-up time, the clocks of the CPU and peripheral will be restored, the SUSLO register and PD bit in PCON register will be cleared by hardware. Program execution resumes with the interrupt service routine. After completion of the interrupt service routine, the instructions which jumped to enter Power-Down mode will continue to run.

(3) Reset signal (logic low on the RESET pin, WDT RESET if enabled, LVR RESET if enabled). This will restore the clock of the CPU after warm-up time, the SUSLO register and the PD bit in PCON register will be cleared by hardware, finally the SH79F3218 will be reset. And the program will execute from address 0000H. The RAM will keep unchanged and the SFR value might be changed according to different function module.

***Note:** In order to entering Idle/Power-Down, it is necessary to add 3 NOPs after setting IDL/PD bit in PCON.*

8.11.4 Register

Table 8.40 Power Control Register

87H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD	SSTAT	SSTAT1	-	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**SMOD**	**Baud rate double bit**
6	**SSTAT**	**SCON[7:5] function selection bit**
5	**SSTAT1**	**SCON1[7:5] function selection bit**
3-2	**GF[1:0]**	**General purpose flags for software use**
1	**PD**	**Power-Down mode control bit** 0: Cleared by hardware when an interrupt or reset occurs 1: Set by software to activate the Power-Down mode
0	**IDL**	**Idle mode control bit** 0: Cleared by hardware when an interrupt or reset occurs 1: Set by software to activate the Idle mode

Table 8.41 Suspend Mode Control Register

8EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SUSLO	SUSLO.7	SUSLO.6	SUSLO.5	SUSLO.4	SUSLO.3	SUSLO.2	SUSLO.1	SUSLO.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SUSLO[7:0]**	This register is used to control the CPU enter suspend mode (Idle or Power-Down). Only consecutive instructions like below will make CPU enter suspend mode. Other wise the either SUSLO, IDL or PD bit will be cleared by hardware in the next machine cycle.

Example

```
IDLE_MODE:
        MOV     SUSLO, #55H
        ORL     PCON, #01H
        NOP
        NOP
        NOP
```

```
POWERDOWN_MODE:
        MOV     SUSLO, #55H
        ORL     PCON, #02H
        NOP
        NOP
        NOP
```

8.12 Warm-up Timer

8.12.1 Feature

- Built-in power on warm-up counter to eliminate unstable state of power on
- Built-in oscillator warm-up counter to eliminate unstable state when oscillation start up

SH79F3218 has a built-in power warm-up counter; it is designed to eliminate unstable state after power on or to do some internal initial operation such as read internal customer code option etc.

SH79F3218 has also a built-in oscillator warm-up counter, it is designed to eliminate unstable state when oscillator starts oscillating in the following conditions: Power-on reset, Pin reset, LVR reset, Watchdog Reset and Wake up from low power consumption mode.

After power-on, SH79F3218 will start power warm-up procedure first, and then oscillator warm-up procedure. Began to run the program after the overflow.

Power Warm-up Time

Power On Reset/ Pin Reset/ Low Voltage Reset		**WDT Reset (Not in Power-Down Mode)**		**WDT Reset (Wakeup from Power-Down Mode)**		**Wakeup from Power-Down Mode (Only for interrupt)**	
TPWRT**	OSC Warm up*	TPWRT**	OSC Warm up*	TPWRT**	OSC Warm up*	TPWRT**	OSC Warm up*
11ms	YES	≈1ms	NO	≈1ms	YES	≈500us	YES

OSC Warm-up Time

Option: OP_WMT **Oscillator Type**	**00**	**01**	**10**	**11**
Ceramic/Crystal	2^{17} X Tosc	2^{14} X Tosc	2^{11} X Tosc	2^{8} X Tosc
32kHz crystal	2^{13} X Tosc			
Internal RC	2^{7} X Tosc			

8.13 CRC verification module

8.13.1 Feature

- Generate CRC check code of the Flash Rom Code, verify the Flash Rom Code whether changed or not
- CRC generator polynomial adopt the CRC-CCITT Standard: $X^{16}+X^{12}+X^{5}+1$, high bit first
- Two mode: High speed CRC mode and Normal CRC speed mode

To improve the system reliability, the SH79F3218 has one CRC verification module built-in, CRC check code can be used to generate real-time code, using the generation polynomial: $X^{16}+X^{12}+X^{5}+1$, which adopt the CRC-CCITT Standard. Users can use this check code compared with the theory value, whether the changes in Flash content monitoring. The last two byte can be stored in the ROM region of the CRC theoretical value (not involved in CRC check), or other location (such as class EEPROM region, sequence number area user identification code area, etc.)

Set CRCADR[3:0] bits can select the CRC check size, set CRC_GO bit to 1 to enable CRC module. After CRC check is done, CRC_GO will be cleared automatically by hardware, and set CRCIF to 1, if interrupt enable bit SCM_LPD_CRC and ECRC are both set to 1, the CRC interrupt will generated in CPU, and the interrupt flag CRCIF will be cleared by software.

Normal CRC mode: the time of CPU operating is not influenced by the CRC check operating, but the time of CRC check is long and uncontrollable.

High speed CRC mode: in order to improve the time of CRC check, there has a way which make CPU into IDLE mode, the time of CRC check will be reduced, CRC interrupt can wake up IDLE mode.

Note: *In Power-Down mode, and CRC is operating, this can make the CRC check code incorrectness. So please make sure system not in the Power-Down mode before CRC is done.*

8.13.2 Register

Table 8.42 CRC Control Register

ACH，Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CRCCON	CRC_GO	CRCIF	-	-	CRCADR3	CRCADR2	CRCADR1	CRCADR0
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	CRC_GO	**CRC GO_DONE Control bit** 0: disable CRC module 1: enable CRC module, cleared when CRC check is done
6	CRCIF	**CRC interrupt flag bit** 0: CRC check is not done, clear by software 1: CRC check is done, set by hardware
3-0	CRCADR[3:0]	**CRC Check address bits** 0000: Check address is 0000 - 07FDH (2K - 2Byte) 0001: Check address is 0000 - 0FFDH (4K - 2Byte) ………… 1110: Check address is 0000 - 77FDH (30K - 2Byte) 1111: Check address is 0000 - 7FFDH (32K - 2Byte)

Table 8.43 CRC Check Data Register

ADH, AEH, Bank0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CRCDL (ADH, Bank0)	CRCD7	CRCD6	CRCD5	CRCD4	CRCD3	CRCD2	CRCD1	CRCD0
CRCDH (AEH, Bank0)	CRCD15	CRCD14	CRCD13	CRCD12	CRCD11	CRCD10	CRCD9	CRCD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**CRCD[15:0]**	**CRC check data register** CRC Check Code updated when CRC Check is completed

Note:

(1) Need a initial value in CRC check data register before CRC enable.

(2) Can not set CRCADR[3:0] bits and CRCDL/CRCDH when CRC is operating.

(3) The last two bytes in the CRC check region is dropped out of CRC check, so can store the CRC check theoretical value.

8.14 Code Option

OP_WDT:

0: Enable WDT function (default)
1: Disable WDT function

OP_WDTPD:

0: Disable WDT function in Power-Down mode (default)
1: Enable WDT function in Power-Down mode

OP_LVREN:

0: Disable LVR function (default)
1: Enable LVR function

OP_LVRLE:

00: 3.7V LVR level 1 (default)
01: 3.1V LVR level 2
10: 2.8V LVR level 3

OP_OSC:

0000: Oscillator1 is internal 16.6M RC, oscillator2 is disabled (default)
0011: Oscillator1 is internal 128k RC, oscillator2 is internal 16.6M RC
0110: Oscillator1 is internal 128k RC, oscillator2 is 2M-16M crystal/ceramic oscillator
1010: Oscillator1 is 32.768k crystal oscillator, oscillator2 is internal 16.6M RC
1110: Oscillator1 is 2M-16M crystal/ceramic oscillator, oscillator2 is disabled

OP_SEG/IO: (The option selects the output current for all modes of application)

00: P0.5-P0.6, P1.2, P1.7, P2.0-P2.6, P3.0-P3.1, When V_{DD}-3V, output 40mA. When GND+3V, Input 40Ma (default)
01: P0.5-P0.6, P1.2, P1.7, P2.0-P2.6, P3.0-P3.1, When V_{DD}-3V, output 30mA. When GND+3V, Input 30mA
10: P0.5-P0.6, P1.2, P1.7, P2.0-P2.6, P3.0-P3.1, When V_{DD}-3V, output 20mA. When GND+3V, Input 20mA
11: P0.5-P0.6, P1.2, P1.7, P2.0-P2.6, P3.0-P3.1, When V_{DD}-3V, output 10mA. When GND+3V, Input 10mA

OP_MODSW:

1: when MODSW = 1, LCD/LED count timer stop work, keep the common value, when LCD/LED enable again, continue scan present Common (default)
0: when MODSW = 1, LCD/LED count timer continue work

OP_WMT: (unavailable for Internal RC)

00: longest warm up time (default)
01: longer warm up time
10: shorter warm up time
11: shortest warm up time

OP_COM/IO: (The option selects the output current for all modes of application)

0: P0.7, P1.0-P1.2, P2.7, P3.0-P3.1, sink ability large mode
1: P0.7, P1.0-P1.2, P2.7, P3.0-P3.1, sink ability normal mode (default)

OP_P0: (The option selects the output current for all modes of application)

0: P0.1-P0.4, sink ability normal mode (default)
1: P0.1-P0.4, sink ability large mode

OP_SCM:

0: SCM is invalid in warm up period (default)
1: SCM is valid in warm up period

OP_OSCDRIVE1:

0: OP_OSCDRIVER2 = 10: 2M Ceramic
OP_OSCDRIVER2 = 11: 4M Ceramic
1: OP_OSCDRIVER2 = 10: 8M Ceramic
OP_OSCDRIVER2 = 11: 12M Ceramic
OP_OSCDRIVER2 = 00: 4M Crystal
OP_OSCDRIVER2 = 01: 8M-12M Crystal (default)

OP_OSCDRIVE2:

10: OP_OSCDRIVE1 = 0: 2M Ceramic
    OP_OSCDRIVE1 = 1: 8M Ceramic
11: OP_OSCDRIVE1 = 0: 4M Ceramic
    OP_OSCDRIVE1 = 1: 12M Ceramic
00: OP_OSCDRIVE1 = 1: 4M Crystal
01: OP_OSCDRIVE1 = 1: 8M-12M Crystal (default)

When using an external vibration ceramic or crystals (OP_OSC = 0110 or 1110)

	OP_OSCDRIVE2: 00	OP_OSCDRIVE2: 01	OP_OSCDRIVE2: 10	OP_OSCDRIVE2: 11
OP_OSCDRIVE1: 0	-	-	2M Ceramic	4M Ceramic
OP_OSCDRIVE1: 1	4M Crystal	8M-12M Crystal	8M Ceramic	12M Ceramic

9. Instruction Set

ARITHMETIC OPERATIONS				
Opcode	**Description**	**Code**	**Byte**	**Cycle**
ADD A, Rn	Add register to accumulator	0x28-0x2F	1	1
ADD A, direct	Add direct byte to accumulator	0x25	2	2
ADD A, @Ri	Add indirect RAM to accumulator	0x26-0x27	1	2
ADD A, #data	Add immediate data to accumulator	0x24	2	2
ADDC A, Rn	Add register to accumulator with carry flag	0x38-0x3F	1	1
ADDC A, direct	Add direct byte to A with carry flag	0x35	2	2
ADDC A, @Ri	Add indirect RAM to A with carry flag	0x36-0x37	1	2
ADDC A, #data	Add immediate data to A with carry flag	0x34	2	2
SUBB A, Rn	Subtract register from A with borrow	0x98-0x9F	1	1
SUBB A, direct	Subtract direct byte from A with borrow	0x95	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	0x96-0x97	1	2
SUBB A, #data	Subtract immediate data from A with borrow	0x94	2	2
INC A	Increment accumulator	0x04	1	1
INC Rn	Increment register	0x08-0x0F	1	2
INC direct	Increment direct byte	0x05	2	3
INC @Ri	Increment indirect RAM	0x06-0x07	1	3
DEC A	Decrement accumulator	0x14	1	1
DEC Rn	Decrement register	0x18-0x1F	1	2
DEC direct	Decrement direct byte	0x15	2	3
DEC @Ri	Decrement indirect RAM	0x16-0x17	1	3
INC DPTR	Increment data pointer	0xA3	1	4
MUL AB 8 X 8 16 X 8	Multiply A and B	0xA4	1	11 20
DIV AB 8 / 8 16 / 8	Divide A by B	0x84	1	11 20
DA A	Decimal adjust accumulator	0xD4	1	1

LOGIC OPERATIONS				
Opcode	**Description**	**Code**	**Byte**	**Cycle**
ANL A, Rn	AND register to accumulator	0x58-0x5F	1	1
ANL A, direct	AND direct byte to accumulator	0x55	2	2
ANL A, @Ri	AND indirect RAM to accumulator	0x56-0x57	1	2
ANL A, #data	AND immediate data to accumulator	0x54	2	2
ANL direct, A	AND accumulator to direct byte	0x52	2	3
ANL direct, #data	AND immediate data to direct byte	0x53	3	3
ORL A, Rn	OR register to accumulator	0x48-0x4F	1	1
ORL A, direct	OR direct byte to accumulator	0x45	2	2
ORL A, @Ri	OR indirect RAM to accumulator	0x46-0x47	1	2
ORL A, #data	OR immediate data to accumulator	0x44	2	2
ORL direct, A	OR accumulator to direct byte	0x42	2	3
ORL direct, #data	OR immediate data to direct byte	0x43	3	3
XRL A, Rn	Exclusive OR register to accumulator	0x68-0x6F	1	1
XRL A, direct	Exclusive OR direct byte to accumulator	0x65	2	2
XRL A, @Ri	Exclusive OR indirect RAM to accumulator	0x66-0x67	1	2
XRL A, #data	Exclusive OR immediate data to accumulator	0x64	2	2
XRL direct, A	Exclusive OR accumulator to direct byte	0x62	2	3
XRL direct, #data	Exclusive OR immediate data to direct byte	0x63	3	3
CLR A	Clear accumulator	0xE4	1	1
CPL A	Complement accumulator	0xF4	1	1
RL A	Rotate accumulator left	0x23	1	1
RLC A	Rotate accumulator left through carry	0x33	1	1
RR A	Rotate accumulator right	0x03	1	1
RRC A	Rotate accumulator right through carry	0x13	1	1
SWAP A	Swap nibbles within the accumulator	0xC4	1	4

DATA TRANSFERS				
Opcode	**Description**	**Code**	**Byte**	**Cycle**
MOV A, Rn	Move register to accumulator	0xE8-0xEF	1	1
MOV A, direct	Move direct byte to accumulator	0xE5	2	2
MOV A, @Ri	Move indirect RAM to accumulator	0xE6-0xE7	1	2
MOV A, #data	Move immediate data to accumulator	0x74	2	2
MOV Rn, A	Move accumulator to register	0xF8-0xFF	1	2
MOV Rn, direct	Move direct byte to register	0xA8-0xAF	2	3
MOV Rn, #data	Move immediate data to register	0x78-0x7F	2	2
MOV direct, A	Move accumulator to direct byte	0xF5	2	2
MOV direct, Rn	Move register to direct byte	0x88-0x8F	2	2
MOV direct1, direct2	Move direct byte to direct byte	0x85	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	0x86-0x87	2	3
MOV direct, #data	Move immediate data to direct byte	0x75	3	3
MOV @Ri, A	Move accumulator to indirect RAM	0xF6-0xF7	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	0xA6-0xA7	2	3
MOV @Ri, #data	Move immediate data to indirect RAM	0x76-0x77	2	2
MOV DPTR, #data16	Load data pointer with a 16-bit constant	0x90	3	3
MOVC A, @A+DPTR	Move code byte relative to DPTR to A	0x93	1	7
MOVC A, @A+PC	Move code byte relative to PC to A	0x83	1	8
MOVX A, @Ri	Move external RAM (8-bit address) to A	0xE2-0xE3	1	5
MOVX A, @DPTR	Move external RAM (16-bit address) to A	0xE0	1	6
MOVX @Ri, A	Move A to external RAM (8-bit address)	0xF2-F3	1	4
MOVX @DPTR, A	Move A to external RAM (16-bit address)	0xF0	1	5
PUSH direct	Push direct byte onto stack	0xC0	2	5
POP direct	Pop direct byte from stack	0xD0	2	4
XCH A, Rn	Exchange register with accumulator	0xC8-0xCF	1	3
XCH A, direct	Exchange direct byte with accumulator	0xC5	2	4
XCH A, @Ri	Exchange indirect RAM with accumulator	0xC6-0xC7	1	4
XCHD A, @Ri	Exchange low-order nibble indirect RAM with A	0xD6-0xD7	1	4

PROGRAM BRANCHES

Opcode	Description	Code	Byte	Cycle
ACALL addr11	Absolute subroutine call	0x11-0xF1	2	7
LCALL addr16	Long subroutine call	0x12	3	7
RET	Return from subroutine	0x22	1	8
RETI	Return from interrupt	0x32	1	8
AJMP addr11	Absolute jump	0x01-0xE1	2	4
LJMP addr16	Long jump	0x02	3	5
SJMP rel	Short jump (relative address)	0x80	2	4
JMP @A+DPTR	Jump indirect relative to the DPTR	0x73	1	6
JZ rel (not taken) (taken)	Jump if accumulator is zero	0x60	2	3 5
JNZ rel (not taken) (taken)	Jump if accumulator is not zero	0x70	2	3 5
JC rel (not taken) (taken)	Jump if carry flag is set	0x40	2	2 4
JNC rel (not taken) (taken)	Jump if carry flag is not set	0x50	2	2 4
JB bit, rel (not taken) (taken)	Jump if direct bit is set	0x20	3	4 6
JNB bit, rel (not taken) (taken)	Jump if direct bit is not set	0x30	3	4 6
JBC bit, rel (not taken) (taken)	Jump if direct bit is set and clear bit	0x10	3	4 6
CJNE A, direct, rel (not taken) (taken)	Compare direct byte to A and jump if not equal	0xB5	3	4 6
CJNE A, #data, rel (not taken) (taken)	Compare immediate to A and jump if not equal	0xB4	3	4 6
CJNE Rn, #data, rel (not taken) (taken)	Compare immediate to reg. and jump if not equal	0xB8-0xBF	3	4 6
CJNE @Ri, #data, rel (not taken) (taken)	Compare immediate to Ri and jump if not equal	0xB6-0xB7	3	4 6
DJNZ Rn, rel (not taken) (taken)	Decrement register and jump if not zero	0xD8-0xDF	2	3 5
DJNZ direct, rel (not taken) (taken)	Decrement direct byte and jump if not zero	0xD5	3	4 6
NOP	No operation	0	1	1

BOOLEAN MANIPULATION				
Opcode	**Description**	**Code**	**Byte**	**Cycle**
CLR C	Clear carry flag	0xC3	1	1
CLR bit	Clear direct bit	0xC2	2	3
SETB C	Set carry flag	0xD3	1	1
SETB bit	Set direct bit	0xD2	2	3
CPL C	Complement carry flag	0xB3	1	1
CPL bit	Complement direct bit	0xB2	2	3
ANL C, bit	AND direct bit to carry flag	0x82	2	2
ANL C, /bit	AND complement of direct bit to carry	0xB0	2	2
ORL C, bit	OR direct bit to carry flag	0x72	2	2
ORL C, /bit	OR complement of direct bit to carry	0xA0	2	2
MOV C, bit	Move direct bit to carry flag	0xA2	2	2
MOV bit, C	Move carry flag to direct bit	0x92	2	3

10. Electrical Characteristics

Absolute Maximum Ratings*

DC Supply Voltage.	-0.3V to +6.0V
Input/Output Voltage.	GND-0.3V to V_{DD}+0.3V
Operating Ambient Temperature.	-40℃ to +85℃
Storage Temperature.	-55℃ to +125℃
FLASH write/erase operating.	0℃ to +85℃

*Comments

Stresses exceed those listed under "**Absolute Maximum Ratings**" may cause permanent damage to this device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied or intended. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

DC Electrical Characteristics(V_{DD} = 2.7V - 5.5V, GND = 0V, T_A = +25°C, unless otherwise specified)

Parameter	Symbol	Min.	Typ.*	Max.	Unit	Condition
Operating Voltage	V_{DD}	2.7	5.0	5.5	V	400kHz ≤ f_{OSC} ≤ 16.6MHz
Operating Current	I_{OP1}	-	5	10	mA	f_{OSC} = 16.6MHz, V_{DD} = 5.0V. All output pins unload (including all digital input pins unfloating) CPU on (execute NOP instruction), WDT on, all other function block off
	I_{OP2}	-	40	80	μA	f_{OSC} = 128kHz, OSCX off,V_{DD} = 5.0V. All output pins unload (including all digital input pins unfloating) CPU on (execute NOP instruction), LVR off, WDT off, all other function block off
Stand by Current (IDLE)	I_{SB1}	-	3	5	mA	f_{OSC} = 16.6MHz, V_{DD} = 5.0V. All output pins unload, CPU off (IDLE), all digital input pins unfloating. LVR on, WDT off, LCD on, all other function block off
	I_{SB2}	-	20	40	μA	f_{OSC} = 128kHz, OSCX off, V_{DD} = 5.0V. All output pins unload (including all digital input pins unfloating) (IDLE MODE). LVR on, WDT off, CRC off, LCD off, all other function block off
Stand by Current (Power-Down)）	I_{SB3}	-	-	15	μA	Osc off, V_{DD} = 5.0V. All output pins unload (including all digital input pins unfloating) CPU off (Power-Down). LVR on, WDT off, CRC off, LCD off, all other function block off
WDT Current	I_{WDT}	-	1	3	μA	V_{DD} = 5.0V, WDT ON
Input Low Voltage 1	V_{IL1}	GND	-	0.3 X V_{DD}	V	I/O Ports, V_{DD} = 2.7 - 5.5V
Input High Voltage 1	V_{IH1}	0.7 X V_{DD}	-	V_{DD}	V	I/O Ports, V_{DD} = 2.7 - 5.5V
Input Low Voltage 2	V_{IL2}	GND	-	0.2 X V_{DD}	V	T0, T1, T2, T3, T4, INT2/3/4, SCK, T2EX, RXD, V_{DD} = 2.7 - 5.5V
Input High Voltage 2	V_{IH2}	0.8 X V_{DD}	-	V_{DD}	V	T0, T1, T2, T3, T4, INT2/3/4, SCK, T2EX, RXD, V_{DD} = 2.7 - 5.5V
Input Leakage Current	I_{IL}	-1	-	1	μA	Input pin, V_{IN} = V_{DD} or GND
output Leakage Current	I_{OL}	-1	-	1	μA	V_{DD} = 5.0V, V_{OUT} = V_{DD} or GND
Pull-up resistor	R_{PH}	-	30	-	kΩ	V_{DD} = 5.0V, V_{IN} = GND
Output High Voltage 1	V_{OH1}	V_{DD} - 3V	-	-	V	I/O Ports, I_{OH} = 40mA, V_{DD} = 5.0V Select Port drive ability normal mode (Code Option)
Output High Voltage 2	V_{OH2}	V_{DD} - 3V	-	-	V	I/O Ports, I_{OH} = 30mA, V_{DD} = 5.0V Select Port drive ability normal mode (Code Option)

(to be continued)

(continue)

Parameter	Symbol	Min.	Typ.*	Max.	Unit	Condition
Output High Voltage 3	V_{OH3}	V_{DD} - 3V	-	-	V	I/O Ports, I_{OH} = 20mA, V_{DD} = 5.0V Select Port drive ability normal mode (Code Option)
Output High Voltage 4	V_{OH4}	V_{DD} - 3V	-	-	V	I/O Ports, I_{OH} = 10mA, V_{DD} = 5.0V Select Port drive ability normal mode (Code Option)
Output Low Voltage 1	V_{OL1}	-	-	GND + 3V	V	I/O Ports, I_{OH} = 40mA, V_{DD} = 5.0V. Select Port drive ability normal mode (CodeOption)
Output Low Voltage 2	V_{OL2}	-	-	GND + 3V	V	I/O Ports, I_{OH} = 30mA, V_{DD} = 5.0V. Select Port drive ability normal mode (Code Option)
Output Low Voltage 3	V_{OL3}	-	-	GND + 3V	V	I/O Ports, I_{OH} = 20mA, V_{DD} = 5.0V. lect Port drive ability normal mode (Code Option)
Output Low Voltage 4	V_{OL4}	-	-	GND + 3V	V	I/O Ports, I_{OH} = 10mA, V_{DD} = 5.0V. lect Port drive ability normal mode (Code Option)
large drive port sink current capability	I_{OL1}	150	180	-	mA	V_{DD} = 5.0V, V_{OL} = 1.0V (COM1-COM7) (OP_COM) Select Port drive ability large mode (CodeOption)
large drive port output current capability	I_{OH1}	-150	-180	-	mA	V_{DD} = 5.0V, V_{OH} = 4.0V (COM1-COM7) (OP_COM) Select Port drive ability large mode (CodeOption)
large drive port sink current capability	I_{OL2}	120	140	-	mA	V_{DD} = 5.0V, V_{OL} = 1.5V (P0.1-P0.4) (OP_P0) Select Port drive ability large mode (CodeOption)

Note:

(1) "" Indicates that the typical value is measured at 5.0V, 25°C, unless otherwise noted.*
(2) maximum current value flowing through V_{DD} 5.0V, 25°C must be less than 200mA.
(3) maximum current value flowing through GND 5.0V, 25°C must be less than 200mA.

AC Electrical Characteristics (V_{DD} = 2.7V - 5.5V, GND = 0V, T_A = +25°C, f_{OSC} = 16.6MHz, unless otherwise specified)

Parameter	Symbol	Min.	Typ.*	Max.	Unit	Condition
RC Frequency	F_{RC}	-	16.6	-	MHz	V_{DD} = 5V
Frequency Stability (RC)	|ΔF|/F	-	-	1	%	RC oscillator: |F-16.6MHz|/16.6MHz (V_{DD} = 2.7-5.5V, T_A = +25℃)
		-	-	2	%	RC oscillator: |F-16.6MHz|/16.6MHz (V_{DD}= 2.7-5.5V, T_A = -40°C~85°C)
		-	128	-	KHz	V_{DD} = 5V
		-	-	5	%	RC oscillator: |F-128kHz|/128kHz (V_{DD} = 2.7-5.5V, T_A = +25℃)
		-10	-	+10	%	RC oscillator: |F-128kHz|/128kHz (V_{DD}= 2.7-5.5V, T_A = -40°C~85°C)

Touch Key Electrical Characteristics (V_{DD} = 2.7V - 5.5V, GND = 0V, T_A = +25°C, unless otherwise specified)

Parameter	Symbol	Min.	Typ.*	Max.	Unit	Condition
Operating Voltage	V_{DD}	2.7	-	5.5	V	
Output Voltage1	V_{DDR1}	3.9	4	4.1	V	V_{DD} = 4.5 - 5.5V
Output Voltage 2	V_{DDR2}	2.9	3.0	3.1	V	IV_{DDR2} = 0 - 20mA, V_{DD} = 3.5 - 5.5V
Output Voltage 3	V_{DDR3}	2.4	2.5	2.6	V	IV_{DDR3} = 0 - 20mA, V_{DD} = 3.0 - 5.5V
Output Voltage 4	V_{DDR4}	1.9	2	2.1	V	IV_{DDR4} = 0 - 20mA, V_{DD} = 2.7 - 5.5V
Output Reference Voltage 1	V_{REF1}	0.9	1	1.1	V	V_{DD} = 2.7 - 5.5V
Output Reference Voltage 2	V_{REF2}	1.4	1.5	1.6	V	V_{DD} = 2.7 - 5.5V
Output Reference Voltage 3	V_{REF3}	1.9	2.0	2.1	V	V_{DD} = 2.7 - 5.5V
Output Reference Voltage 4	V_{REF4}	2.4	2.5	2.6	V	V_{DD} = 3.0 - 5.5V

Analog Comparator Electrical Characteristics (V_{DD} = 2.7 - 5.5V, GND = 0V, T_A = +25°C, unless otherwise specified)

Parameter	Symbol	Min.	Typ.*	Max.	Unit	Condition
Input Offset Voltage	VIO	-	10	15	mV	T_A = 25°C
Offset voltage of Input Voltage with temperature changes	VI1	-	3	5	mV	T_A = -40°C~85°C
Input Common-Mode Voltage Range	VICM	0	-	V_{DD}-1V	V	

Low Voltage Reset Electrical Characteristics (V_{DD} = 2.7V - 5.5V, GND = 0V, T_A = +25°C, unless otherwise specified)

Parameter	Symbol	Min.	Typ.*	Max.	Unit	Condition
LVR Voltage 1	V_{LVR}	3.0	3.1	3.2	V	LVR enabled
LVR Voltage 2	V_{LVR}	3.6	3.7	3.8	V	LVR enabled
LVR Voltage 3	V_{LVR}	2.7	2.8	2.9	V	LVR enabled
Drop-Down Pulse Width for LVR	T_{LVR}	-	30	-	μs	

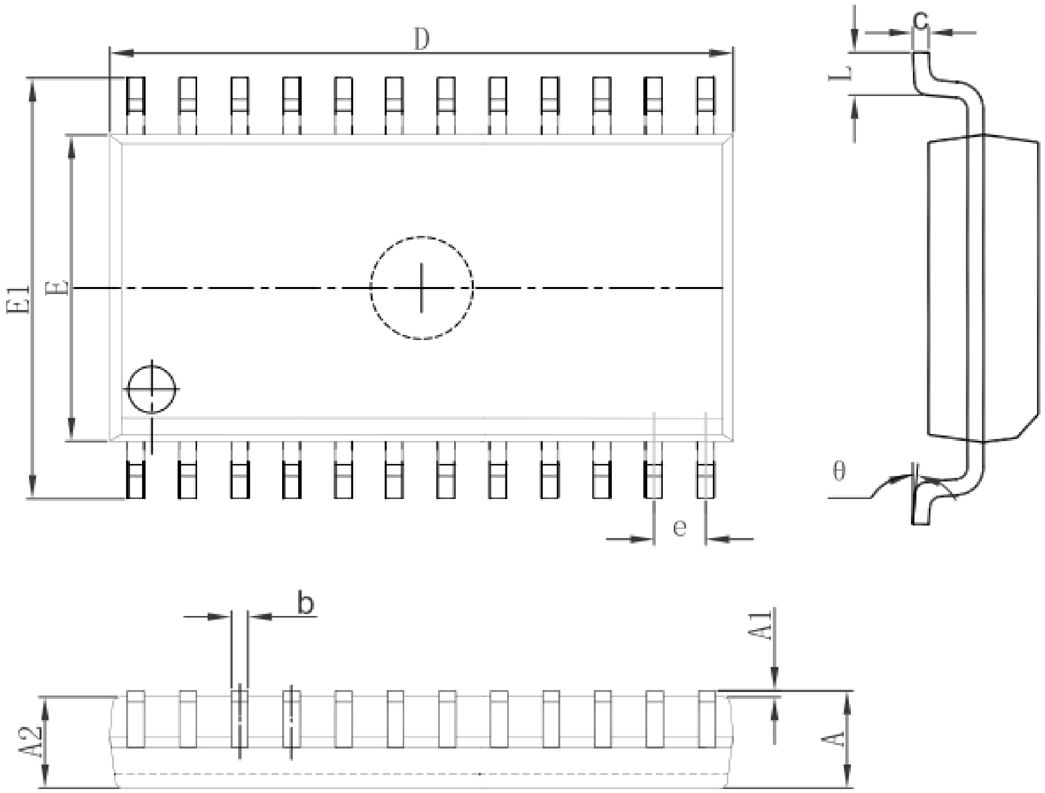
11. Ordering Information

Part No.	Package
SH79F3218M/024MU	24 SOP

12. Package Information

SOP24L Outline Dimensions

unit: inches/mm



Symbol	Dimensions in inches		Dimensions in mm	
	Min	Max	Min	Max
A	2.350	2.650	0.093	0.104
A1	0.100	0.300	0.004	0.012
A2	2.100	2.500	0.083	0.098
b	0.330	0.510	0.013	0.020
c	0.204	0.330	0.008	0.013
D	15.200	15.600	0.598	0.614
E	7.400	7.600	0.291	0.299
E1	10.210	10.610	0.402	0.418
e	1.270 (BSC)		0.050 (BSC)	
L	0.400	1.270	0.016	0.050
θ	0°	8°	0°	8°

13. Product SPEC. Change Notice

Version	Content	Date
1.0	Original	Nov. 2016

Content