SINO WEALTH

SH79F1612B

Enhanced 8051 Microcontroller with 10bit ADC

1. Features

- 8bits micro-controller with Pipe-line structured 8051 compatible instruction set
- Flash ROM: 16K Bytes
- RAM: internal 256 Bytes, external 768 Bytes
- EEPROM-like: 512 Bytes
- Operation Voltage:
  f_{OSC} = 30kHz - 16.6MHz, V_{DD} = 2.8V - 5.5V
- Oscillator (code option)
  - Crystal oscillator: 32.768kHz
  - Crystal oscillator: 4MHz - 16.6MHz
  - Ceramic oscillator: 2MHz -16.6MHz
  - Internal RC: 16.6MHz (±2%)
  - External clock: 30kHz - 16.6MHz
- 14/18 CMOS bi-directional I/O pins
- 5/7 high current drive ports
- Three 16-bit timer/counters T0, T1 and T2
- Powerful interrupt sources:
  - Timer0, 1, 2
  - INT0, 1, 2
  - ADC, EUART, SCM, PWM, LPD, EUART
- One 8-bit PWM
- 1/2 EUART
- Low voltage detect (LPD)
- 6/8 channels 10-bits Analog Digital Converter (ADC), with comparator function built-in
- Low Voltage Reset (LVR) function (enabled by code option)
  - LVR voltage level 1: 4.1V
  - LVR voltage level 2: 3.7V
  - LVR voltage level 3: 2.8V
- CPU Machine cycle: 1 oscillator clock
- Watch Dog Timer (WDT)
- Warm-up Timer
- System Clock Monitor (SCM)
- Support Low power operation modes
  - Idle Mode
  - Power-Down Mode
- Low power consumption
- Flash Type
- Package:
  - SOP/TSSOP 20Pin
  - SOP 16Pin

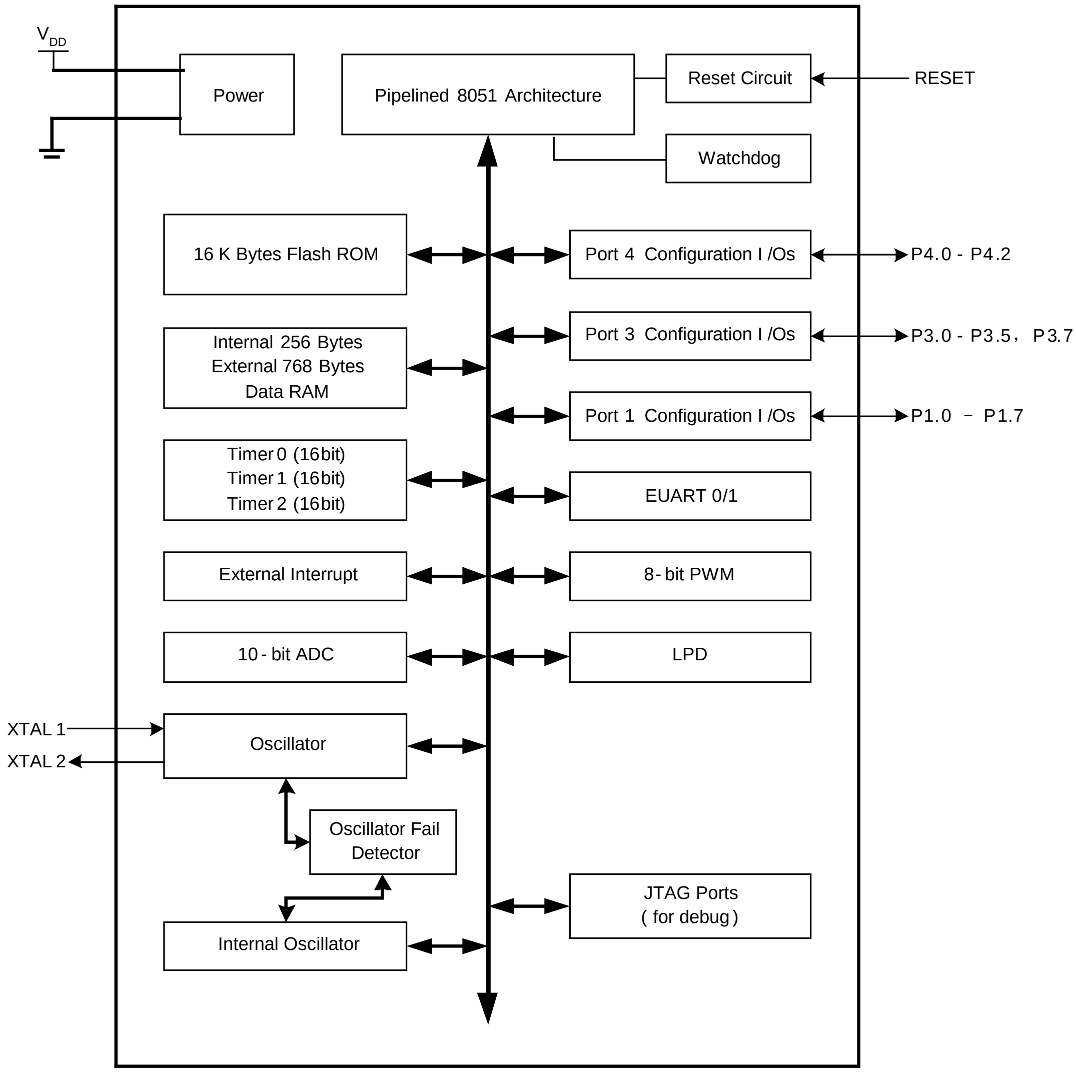
2. General Description

The SH79F1612B is a high performance 8051 compatible micro-controller, regard to its build-in Pipe-line instruction fetch structure, that helps the SH79F1612B can perform more fast operation speed and higher calculation performance, if compare SH79F1612B with standard 8051 at same clock speed.

The SH79F1612B retains most features of the standard 8051. These features include internal 256 bytes RAM, 1/2 UART and INT0-2.In addition, the SH79F1612B provides external 768 bytes RAM, It also contains 16K bytes Flash memory block both for program and data. Also the ADC and PWM timer functions are incorporated in SH79F1612B.

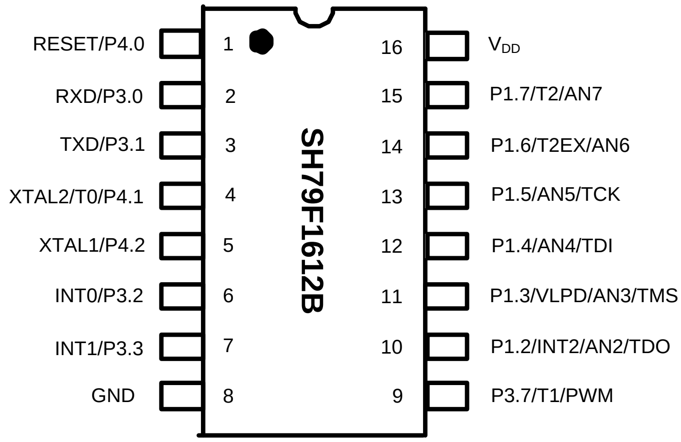
For high reliability and low cost issues, the SH79F1612B builds in Watchdog Timer, Low Voltage Reset function. And SH79F1612B also supports two power saving modes to reduce power consumption.

3. Block Diagram

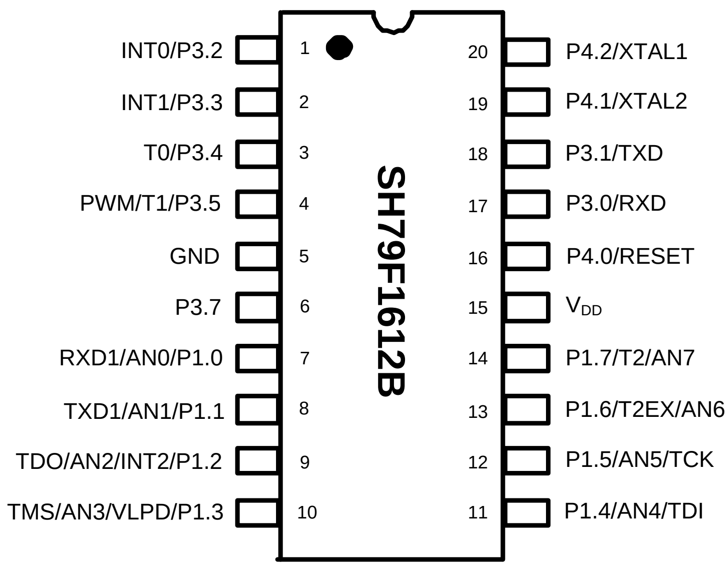


4. Pin Configuration

4.1 SOP 16Pin



4.2 SOP/TSSOP 20Pin



Pin Configuration Diagram

Notice: *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 SOP 16	Pin Name	Default Function
1	P4.0/RESET	RESET
2	RXD/P3.0	P3.0
3	TXD/P3.1	P3.1
4	P4.1/T0/XTAL2	P4.1 or Oscillator output
5	P4.2/XTAL1	P4.2 or Oscillator input
6	INT0/P3.2	P3.2
7	INT1/P3.3	P3.3
8	GND	-----
9	PWM/T1/P3.7	P3.7
10	TDO/AN2/INT2/P1.2	P1.2
11	TMS/AN3/VLPD/P1.3	P1.3
12	TDI/AN4/P1.4	P1.4
13	TCK/AN5/P1.5	P1.5
14	AN6/T2EX/P1.6	P1.6
15	AN7/T2/P1.7	P1.7
16	V_{DD}	-----

Pin No SOP/TSSOP20	Pin Name	Default Function
1	INT0/P3.2	P3.2
2	INT1/P3.3	P3.3
3	T0/P3.4	P3.4
4	PWM/T1/P3.5	P3.5
5	GND	-----
6	P3.7	P3.7
7	RXD1/AN0/P1.0	P1.0
8	TXD1/AN1/P1.1	P1.1
9	TDO/AN2/INT2/P1.2	P1.2
10	TMS/AN3/VLPD/P1.3	P1.3
11	TDI/AN4/P1.4	P1.4
12	TCK/AN5/P1.5	P1.5
13	AN6/T2EX/P1.6	P1.6
14	AN7/T2/P1.7	P1.7
15	V_{DD}	-----
16	P4.0/RESET	RESET
17	RXD/P3.0	P3.0
18	TXD/P3.1	P3.1
19	P4.1/XTAL2	P4.1 or Oscillator output
20	P4.2/XTAL1	P4.2 or Oscillator input

5. Pin Description

Pin No.	Type	Description
I/O PORT		
P4.0 - P4.2	I/O	3 bit General purpose CMOS I/O
P3.0 - P3.7	I/O	7 bit General purpose CMOS I/O
P1.0 - P1.7	I/O	8 bit General purpose CMOS I/O
Timer		
T0	I/O	Timer0 external input/Comparator output
T1	I/O	Timer1 external input/Comparator output
T2	I/O	Timer2 external input/Baud-Rate generator
T2EX	I	Timer2 Reload/Capture/Direction Control
PWM		
PWM	O	Output pin for 8-bit PWM timer
EUART		
RXD	I	EUART data input
TXD	O	EUART data output
RXD1	I	EUART1 data input
TXD1	O	EUART1 data output
ADC		
AN0 - AN7	I	ADC input channel
Interrupt & Reset & Clock & Power		
INT0 - INT2	I	External interrupt 0-2 input source
RESET	I	Reset pin (Logic high reset)
XTAL1	I	Oscillator input
XTAL2	O	Oscillator output
V_{DD}	P	Power supply (2.8 - 5.5V)
GND	P	Ground
VLPD		
VLPD	I	Power voltage detect
Programmer		
TDO	O	Debug interface: Test data out
TMS	I	Debug interface: Test mode select
TDI	I	Debug interface: Test data in
TCK	I	Debug interface: Test clock in
Note: *When P1.2-1.5 used as debug interface, functions of P1.2-1.5 are blocked.*		

6. SFR Mapping

The SH79F1612B provides 256 bytes of internal RAM to contain general-purpose data memory and Special Function Register (SFR). The SFR of the SH79F1612B 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
LPD Register:	LPDCON
Flash Registers:	IB_OFFSET, XPAGE, IB_DATA, IB_CON1, IB_CON2, IB_CON3, IB_CON4, IB_CON5, FLASHCON
Data Memory Register:	XPAGE
System Clock Control Register:	CLKCON
Hardware Watchdog Timer Registers:	RSTSTAT
Interrupt System Registers:	IEN0, IEN1, IPH0, IPL0, IPH1, IPL1, EXF0
I/O Port Registers:	P1, P3, P4, P1M0, P1M1, P3M0, P3M1, P4M0, P4M1
Timer Registers:	TCON, TMOD, TL0, TH0, TL1, TH1, TCON1, T2CON, T2MOD, RCAP2H, RCAP2L
EUART Registers:	SCON, SBUF, SADEN, SADDR, PCON, SCON1, SBUF1, SADEN1, SADDR1, SBRTH1, SBRTL1, SFINF1, PCON1
ADC Registers:	ADCON, ADT, ADCH, ADDL, ADDH
PWM Registers:	PWMCON, PWMP, PWMD

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	AUXC 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 Pointer1 Low byte	00000000	DPL1.7	DPL1.6	DPL1.5	DPL1.4	DPL1.3	DPL1.2	DPL1.1	DPL1.0
DPH1	85H	Data Pointer1 High byte	00000000	DPH1.7	DPH1.6	DPH1.5	DPH1.4	DPH1.3	DPH1.2	DPH1.1	DPH1.0
INSCON	86H	Data pointer select	----00-0	-	-	-	-	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	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	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	Flash Memory Control Register1	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	Flash Memory Control Register2	----0000	-	-	-	-	IB_CON2.3	IB_CON2.2	IB_CON2.1	IB_CON2.0
IB_CON3	F4H	Flash Memory Control Register3	----0000	-	-	-	-	IB_CON3.3	IB_CON3.2	IB_CON3.1	IB_CON3.0
IB_CON4	F5H	Flash Memory Control Register4	----0000	-	-	-	-	IB_CON4.3	IB_CON4.2	IB_CON4.1	IB_CON4.0
IB_CON5	F6H	Flash Memory Control Register5	----0000	-	-	-	-	IB_CON5.3	IB_CON5.2	IB_CON5.1	IB_CON5.0
XPAGE	F7H	Memory Page	00000000	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
FLASHCON	A7H	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	Watchdog Timer Control	*-***000	WDOF	-	PORF	LVRF	CLRF	WDT.2	WDT.1	WDT.0

***Note:** *RSTSTAT initial value is determined by different RESET.*

Table 6.5 CLKCON SFR

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	B2H	System Clock Control Register	111000--	32K_SPDUP	CLKS1	CLKS0	SCMIF	RCON	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	Interrupt Enable Control0	00000000	EA	EADC	ET2	ES0	ET1	EX1	ET0	EX0
IEN1	A9H	Interrupt Enable Control1	0-0000--	ELPD	-	EPWM	ESCM	ES1	EX2	-	-
EXF0	E8H	External interrupt Control0	----00-0	-	-	-	-	IT2.1	IT2.0	-	IE2
IPL0	B8H	Interrupt Priority Control Low0	-0000000	-	PADCL	PT2L	PSL	PT1L	PX1L	PT0L	PX0L
IPH0	B4H	Interrupt Priority Control High0	-0000000	-	PADCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
IPL1	B9H	Interrupt Priority Control Low1	0-0000--	PLPDL	-	PPWML	PSCML	PSL1	PX2L	-	-
IPH1	B5H	Interrupt Priority Control High1	0-0000--	PLPDH	-	PPWMH	PSCMH	PSH1	PX2H	-	-

Table 6.7 Port SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1	90H	8-bit Port1	11111111	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P3	B0H	7-bit Port3	1-111111	P3.7	-	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
P4	C0H	3-bit Port4	-----111	-	-	-	-	-	P4.2	P4.1	P4.0
P1M0	EAH		00000000	P1M07	P1M06	P1M05	P1M04	P1M03	P1M02	P1M01	P1M00
P1M1	E2H		00000000	P1M17	P1M16	P1M15	P1M14	P1M13	P1M12	P1M11	P1M10
P3M0	ECH		0-000000	P3M07	-	P3M05	P3M04	P3M03	P3M02	P3M01	P3M00
P3M1	E4H		0-000000	P3M17	-	P3M15	P3M14	P3M13	P3M12	P3M11	P3M10
P4M0	EDH		-----000	-	-	-	-	-	P4M02	P4M01	P4M00
P4M1	E5H		-----000	-	-	-	-	-	P4M12	P4M11	P4M10

Table 6.8 Timer SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	88H	Timer/Counter0/1 Control	00000000	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	89H	Timer/Counter0/1 Mode	00000000	GATE1	$C/\overline{T1}$	M11	M10	GATE0	$C/\overline{T0}$	M01	M00
TL0	8AH	Timer/Counter0 Low Byte	00000000	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
TH0	8CH	Timer/Counter0 High Byte	00000000	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
TL1	8BH	Timer/Counter1 Low Byte	00000000	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.1
TH1	8DH	Timer/Counter1 High Byte	00000000	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.1
T2CON	C8H	Timer/Counter2 Control	00000000	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	$CP/\overline{RL2}$
T2MOD	C9H	Timer/Counter2 Control	0-----00	TCLKP2	-	-	-	-	-	T2OE	DCEN
RCAP2L	CAH	Timer/Counter2 Reload /Caprure Low Byte	00000000	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
RCAP2H	CBH	Timer/Counter2 Reload /Caprure High Byte	00000000	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
TL2	CCH	Timer/Counter2 Low Byte	00000000	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
TH2	CDH	Timer/Counter2 High Byte	00000000	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
TCON1	CEH	Timer/Counter2 Control	-00-0000	-	TCLKS1	TCLKS0	-	TCLKP1	TCLKP0	TC1	TC0

Table 6.9 EUART SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON	98H	Serial Control	00000000	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI
SBUF	99H	Serial Data Buffer	00000000	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
SADDR	9AH	Slave Address	00000000	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
SADEN	9BH	Slave Address Mask	00000000	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
PCON	87H	Power & serial Control	00--0000	SMOD	SSTAT	-	-	GF1	GF0	PD	IDL
SCON1	D8H	Serial Control	00000000	SM10/FE1	SM11/RXOV1	SM12/TXCOL1	REN1	TB81	RB81	TI1	RI1
SBUF1	D9H	Serial Data Buffer	00000000	SBUF1.7	SBUF1.6	SBUF1.5	SBUF1.4	SBUF1.3	SBUF1.2	SBUF1.1	SBUF1.0
SADDR1	DAH	Slave Address	00000000	SADDR1.7	SADDR1.6	SADDR1.5	SADDR1.4	SADDR1.3	SADDR1.2	SADDR1.1	SADDR1.0
SADEN1	DBH	Slave Address Mask	00000000	SADEN1.7	SADEN1.6	SADEN1.5	SADEN1.4	SADEN1.3	SADEN1.2	SADEN1.1	SADEN1.0
SBRTH1	DCH	Baudrate Generator	00000000	SBRTEN1	SBRT1.14	SBRT1.13	SBRT1.12	SBRT1.11	SBRT1.10	SBRT1.9	SBRT1.8
SBRTL1	DDH	Baudrate Generator	00000000	SBRT1.7	SBRT1.6	SBRT1.5	SBRT1.4	SBRT1.3	SBRT1.2	SBRT1.1	SBRT1.0
SFINE1	DEH	Baudrate Generator	----0000	-	-	-	-	SFINE1.3	SFINE1.2	SFINE1.1	SFINE1.0
PCON1	DFH	Power & serial Control	00------	SMOD1	SSTAT1	-	-	-	-	-	-

Table 6.10 ADC SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON	93H	ADC Control	00000000	ADON	ADCIF	EC	SCH3	SCH2	SCH1	SCH0	GO/$\overline{\text{DONE}}$
ADT	94H	ADC Time Configuration	000-0000	TADC2	TADC1	TADC0	-	TS3	TS2	TS1	TS0
ADCH	95H	ADC Channel Configuration	00000000	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
ADDL	96H	ADC Data Low Byte	------00	-	-	-	-	-	-	A1	A0
ADDH	97H	ADC Data High Byte	00000000	A9	A8	A7	A6	A5	A4	A3	A2

Table 6.11 PWM SFRs

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON	D1H	8 bit PWM control	0000--00	PWMEN	PWMS	PWMCK1	PWMCK0	-	-	PWMIF	PWMSS
PWMP	D2H	8-bit PWM Period Control low byte	00000000	PWMP.7	PWMP.6	PWMP.5	PWMP.4	PWMP.3	PWMP.2	PWMP.1	PWMP.0
PWMD	D3H	8-bit PWM Period Control high byte	00000000	PWMD.7	PWMD.6	PWMD.5	PWMD.4	PWMD.3	PWMD.2	PWMD.1	PWMD.0

Table 6.12 LPD SFR

Mnem	Add	Name	POR/WDT/LVR /PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPDCON	B3H	LPD Control	000--000	LPDEN	LPDF	LPDV	-	-	LPDS2	LPDS1	LPDS0

***Note:**- :Unimplemented*

SFR Map

	Bit addressable	Non Bit addressable							
	0/8	*1/9*	*2/A*	*3/B*	*4/C*	*5/D*	*6/E*	*7/F*	
F8H				IB_OFFSET	IB_DATA				FFH
F0H	B	AUXC	IB_CON1	IB_CON2	IB_CON3	IB_CON4	IB_CON5	XPAGE	F7H
E8H	EXF0		P1M0		P3M0	P4M0			EFH
E0H	ACC		P1M1		P3M1	P4M1			E7H
D8H	SCON1	SBUF1	SADDR1	SADEN1	SBRTH1	SBRTL1	SFINE1	PCON1	DFH
D0H	PSW	PWMCON	PWMP	PWMD					D7H
C8H	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	TCON1		CFH
C0H	P4								C7H
B8H	IPL0	IPL1							BFH
B0H	P3	RSTSTAT	CLKCON	LPDCON	IPH0	IPH1			B7H
A8H	IEN0	IEN1							AFH
A0H								FLASHCON	A7H
98H	SCON	SBUF	SADDR	SADEN					9FH
90H	P1			ADCON	ADT	ADCH	ADDL	ADDH	97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	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. The mnemonics for accumulator-specific instructions, however, refer to the Accumulator simply as A.

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 wide, 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.

Table 7.1 PSW Register

D0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	C	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	**C**	**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: an 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: an even number of “one” bits in the Accumulator 1: an odd number of “one” bits in the Accumulator

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.

7.1.2 Enhanced CPU core SFRs

Feature

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

The SH79F1612B has modified 'MUL' and 'DIV' instructions. These instructions support 16 bit operand. A new register - the register 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 the same with 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. The user can switch data pointers by toggling the DPS bit. 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	-	-	-	-	DIV	MUL	-	DPS
R/W	-	-	-	-	R/W	R/W	-	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	-	0

Bit Number	Bit Mnemonic	Description
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

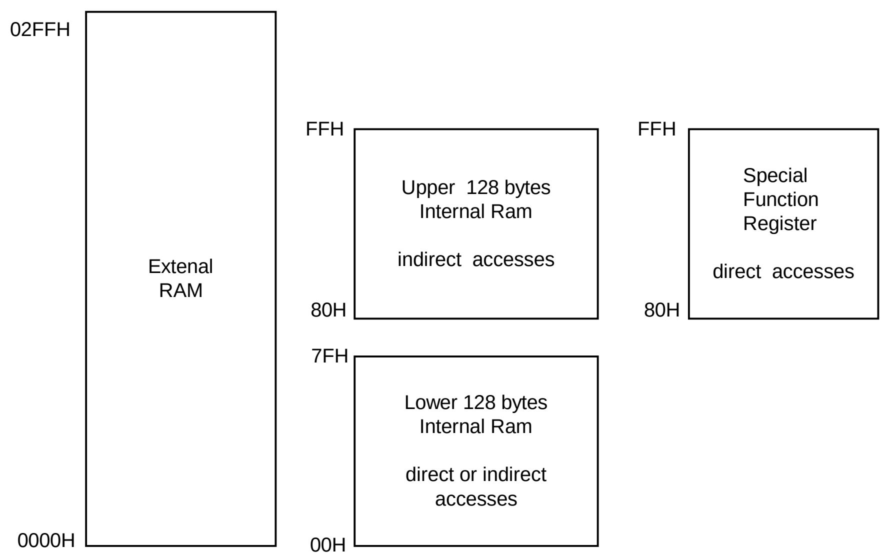
7.2.1 Feature

SH79F1612B provides both internal RAM 256bytes and external RAM 768bytes 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 external 768bytes of RAM(addresses 00H to 2FFH) 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.

Note: *the unused address is unavailable in SFR.*



The Internal and External RAM Configuration

The SH79F1612B 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 768 bytes RAM.

In SH79F1612B 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

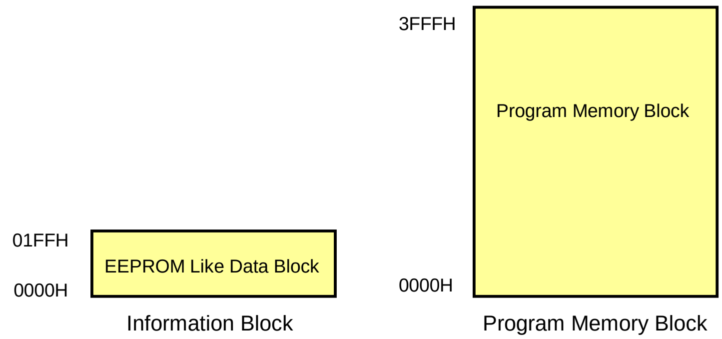
F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	-	-	-	-	-	-	XPAGE.1	XPAGE.0
R/W	-	-	-	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-	-	0	0

Bit Number	Bit Mnemonic	Description
1:0	XPAGE[1:0]	**RAM Page Selector**

7.3 Flash Program Memory

7.3.1 Feature

- The program memory consists 16 X 1KB sectors, total 16KB
- Programming and erase can be done over the full operation voltage range
- Write, read and erase operation are all supported by In-Circuit Programming (ICP)
- Fast mass/sector erase and programming
- Minimum program/erase cycles: 100,000
- Minimum years data retention: 10
- Low power consumption



The SH79F1612B embeds 16K 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. 1024 bytes per sector.

The SH79F1612B also embeds 512bytes EEPROM-like for program data. 256bytes per sector.

Flash option definition:

In-Circuit Programming (ICP) mode: Erase, read and write to flash memory by the Flash Programmer

Self-Sector Programming (SSP) mode: Erase, read and write to flash memory by the user code in program memory

Flash memory supports the following operations:

(1) Code-Protect Control mode Programming

SH79F1612B implements code-protect function to offer high safeguard for customer code. Two modes are available for each sector.

Code-protect control mode 0: Used to enable/disable the write/read operation (except mass erase) from any programmer.

Code-protect control mode 1: Used to enable/disable the read operation through MOVC instruction from other sectors; or the sector erase/write operation through **SSP** Function.

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 Erase

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

The Write/Read Code operation will write the customer code into the Flash Programming Memory or read the customer code from the Flash Programming Memory. This operation can be done by Flash Programmer or the user's program.

If done by the user's program, the code-protect control mode 1 of the selected sector must be disabled. But the program can read/write its own sector regardless of its security bit.

If done by the Flash Programmer, the code-protect control mode 0 of the selected sector must be disabled.

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

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.

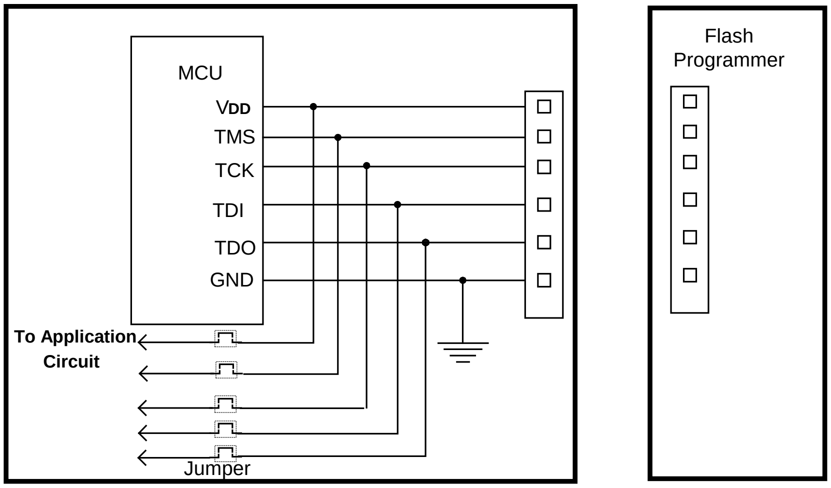
Operation	ICP	SSP
Code Protection	Yes	No
Sector Erase	Yes (without security bit)	Yes (without security bit)
Mass Erase	Yes	No
EEPROM-like Erase	Yes	Yes
Write/Read	Yes (without security bit)	Yes (without security bit or its own sector)
EEPROM-like Write/Read	Yes	Yes

7.3.2 Flash Operation in ICP Mode

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 wires (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 three 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 timing is very sensitive, five jumpers are needed (V_{DD}, TDO, TDI, TCK, TMS) to separate the program pins from the application circuit as the following diagram.



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 and short these jumpers after programming is complete.

7.4 SSP Function

The SH79F1612B provides SSP (Self Sector Programming) function, each sector can be sector erased or programmed by the user's code if the selected sector is not be protected. But once sector has been programmed, it cannot be reprogrammed before sector erase.

The SH79F1612B builds in a complex control flow to prevent the code from carelessly modification. If the dedicated conditions are not met (IB_CON1-5), the SSP will be terminated.

7.4.1 SSP Register

Table 7.4 Offset Register for Programming

F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	-	-	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
Reset Value (POR/WDT/LVR/PIN)	-	-	0	0	0	0	0	0

Flash memory, one sector is 1024 bytes

Bit Number	Bit Mnemonic	Description
5-2	**XPAGE[5:2]**	Sector of the flash memory to be programmed, 0000---means sector 0, and so on
1-0	**XPAGE[1:0]**	High Address of Offset of the flash memory sector to be programmed

EEPROM-like memory, one sector is 256 bytes

Bit Number	Bit Mnemonic	Description
7-1	**XPAGE[7:1]**	Reserved
0	**XPAGE[0]**	Sector of the flash memory to be programmed, 0---means sector 0, and so on

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 Address of Offset of the flash memory sector to be programmed

Table 7.6 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

Table 7.7 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

Table 7.8 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, else Flash Programming will terminate

Table 7.9 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 else Flash Programming will terminate

Table 7.10 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

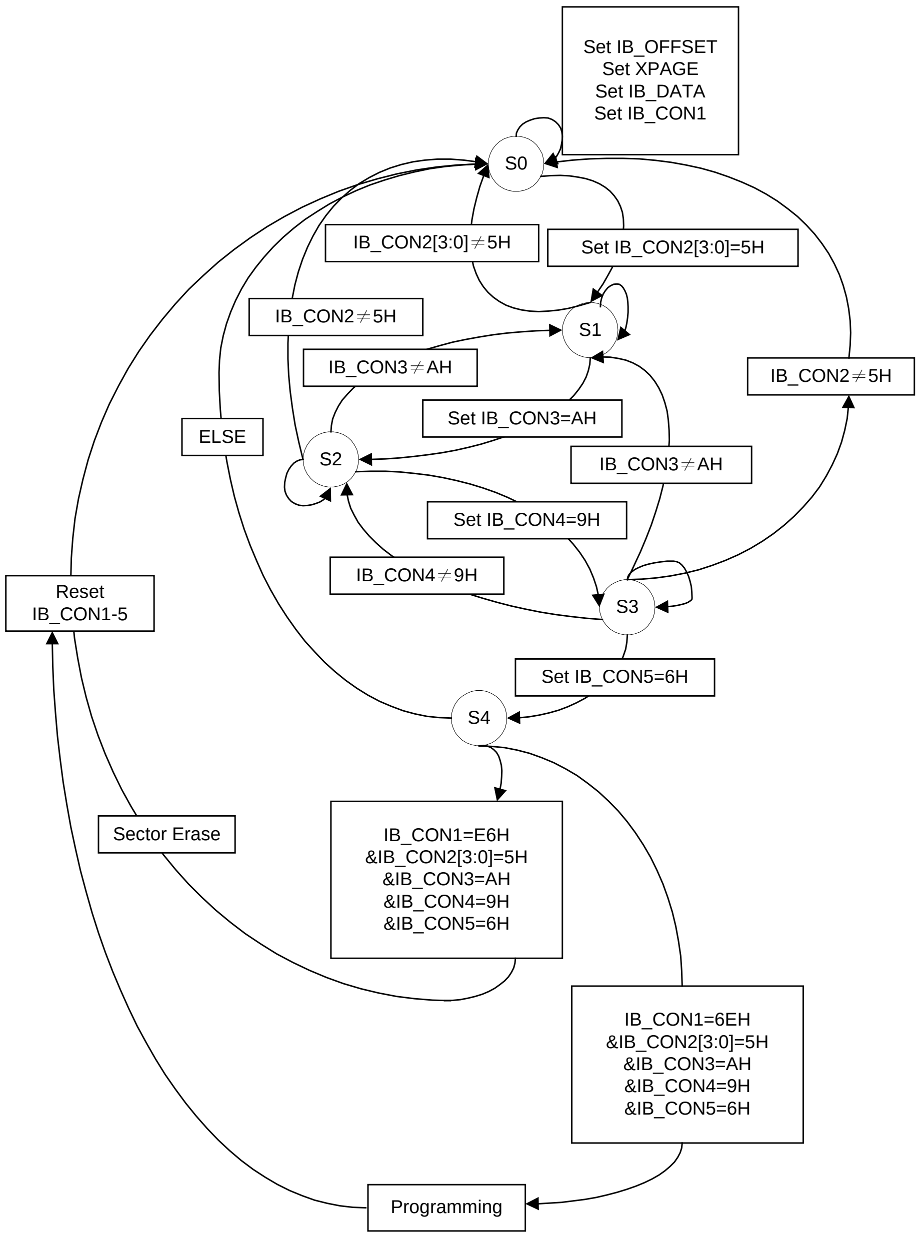
Bit Number	Bit Mnemonic	Description
3-0	**IB_CON4[3:0]**	Must be 09H, else Flash Programming will terminate

Table 7.11 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, else 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 following the steps below:

(1) For Code/Data Programming

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 NOP 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

1. Disable interrupt;
2. Fill in the XPAGE for the corresponding sector;
3. Fill in IB_CON1-5 sequentially;
4. Add 4 NOP 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

Steps is same as code programming,the diffenrences are:
1. Set FAC bit in FLASHCON register before programming or erase EEPROM-Like.
2. One sector of EEPROM-Like is 256 bytes.

7.4.4 Readable Random Code

Every chip is cured an 8-bit readable random code after production. Readable random code is 0-255 random value,and can not be erased, read by program or tools.

How to read random code: set FAC bit, Assigned to the DPTR as "0A7FH", clear A, then use "MOVC A, @A+DPTR" to read.

Table 7.12 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
7-1	**-**	**Reserved**
0	**FAC**	**FAC: Flash access control** 0: MOVC or SSP access main memory 1: MOVC or SSP access EEPROM-like

Note: *After reading random code, users must clear FAC bit, otherwise it will affect the user program the ROM reading instruction program.*

7.5 System Clock and Oscillator

7.5.1 Feature

- 5 oscillator types: 32.768kHz crystal, crystal oscillator, ceramic oscillator, external clock and 16.6MHz internal RC
- Built-in 16.6MHz (±2%) Internal RC
- Built-in 32.768kHz speed up circuit
- Built-in system clock prescaler

7.5.2 Clock Definition

The SH79F1612B have several internal clocks defined as below:

OSCCLK: the oscillator clock from one of the five oscillator types (32.768kHz crystal, crystal oscillator, ceramic oscillator, external clock and interal RC) f_{OSC} is defined as the OSCCLK frequency. t_{OSC} is defined as the OSCCLK period.

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

SYSCLK: system clock, the output 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

SH79F1612B has 5 oscillator types: 32.768kHz crystal, crystal oscillator (4MHz-16.6MHz), ceramic Oscillator (2MHz-16.6MHz), external Oscillator (30kHz-16.6MHz) and internal RC (16.6MHz), which is 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 7.13 System Clock Control Register

B2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	32K_SPDUP	CLKS1	CLKS0	SCMIF	RCON	FS	-	-
R/W	R/W	R/W	R/W	R/W	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 011, 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.

(to be continued)

(continue)

3	**RCON**	**Internal RC On control Register** 0: Turn off Internal RC 1: Turn on Internal RC Only when code option OP_OSC is 011, this bit is valid. (32.768kHz oscillator is selected, Refer to **code option** section for details)
2	**FS**	**Frequency Select Register** 0: 32.768kHz is selected as OSCSCLK 1: Internal RC is selected as OSCSCLK Only when code option OP_OSC is 011. this bit is valid. (32.768kHz oscillator is selected, Refer to **code option** section for details)

Note:

RCON and FS is valid only when code option OP_OSC is 011.When Internal RC is used as OSCSCLK (that is RCON = 1 and FS = 1), RCON is can't be cleared by software. System Clock Monitor function is blocked.

When OSCSCLK changed from 32.768kHz to Internal RC, the steps below must be done in sequence:

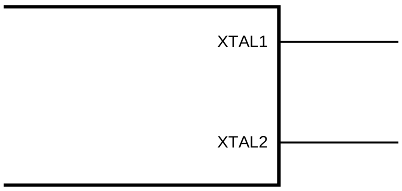
(1) Set RCON = 1 to turn on the Internal RC;

(2) Wait at least 2 Oscillator period;

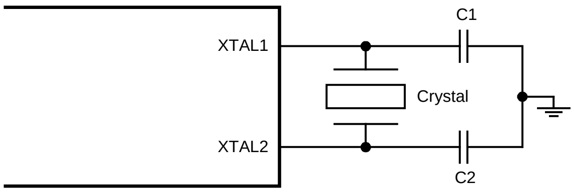
(3) Set FS = 1 to select SYSCLK as Internal RC.

7.5.5 Oscillator Type

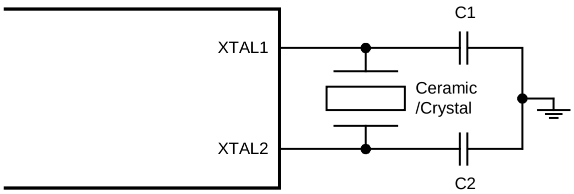
(1) Internal RC: 16.6MHz



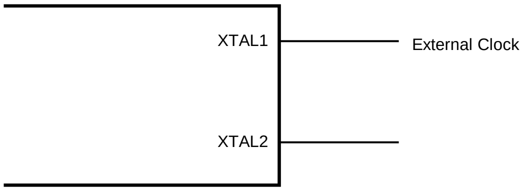
(2) Crystal Oscillator 32.768kHz and internal RC: 16.6MHz



(3) Crystal Oscillator(Ceramic resonator): 4MHz(2MHz) - 16.6MHz



(4) External clock: 30kHz - 16.6MHz



7.5.6 Capacitor Selection for Oscillator

Ceramic Resonators			Remarks
Frequency	**C1**	**C2**	
2MHz	25 - 30pF	25 - 30pF	no bulit-in ceramic resonator load capacitance
4MHz	25 - 30pF	25 - 30pF	Different parameters under model
	12 - 15pF	12 - 15pF	
	15pF	15pF	
6MHz	25 - 30pF	25 - 30pF	Different parameters under model
	12 - 15pF	12 - 15pF	
	15pF	15pF	
8MHz	25 - 30pF	25 - 30pF	Different parameters under model
	7 - 10pF	7 - 10pF	
	12 - 15pF	12 - 15pF	
	15pF	15pF	
10MHz	7 - 10pF	7 - 10pF	Different parameters under model
	15pF	15pF	
12MHz	7 - 10pF	7 - 10pF	No
16MHz	7 - 10pF	7 - 10pF	Different parameters under model
	28pF	28pF	

Crystal Oscillator			Remarks
Frequency	**C1**	**C2**	
32.768kHz	10 - 15pF	10 - 15pF	No
4MHz	10 - 15pF	10 - 15pF	No
8MHz	10 - 15pF	10 - 15pF	No
10MHz	10 - 15pF	10 - 15pF	No
12MHz/16MHz	10 - 15pF	10 - 15pF	No

Notes:

(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.

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, SH79F1612B contains a system clock monitor (SCM) module. If the system clock fails (for example the oscillator stops oscillating), the built-in SCM will switch the OSCCLK to the internal 32k WDTCLK and set system clock monitor bit (SCMIF) to 1. And the SCM interrupt will be generated when EA and ESCM is enabled. If the OSCCLK comes back, SCM will switch the OSCCLK back to the oscillator and clears the SCMIF automatically.

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 fail automatically.

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

Table 7.14 System Clock Control Register

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

Bit Number	Bit Mnemonic	Description
4	**SCMIF**	**System Clock Monitor 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 Feature

- 14/18 bi-directional I/O ports
- Four selectable I/O mode
- Share with alternative functions

The SH79F1612B has 14/18 bi-directional I/O ports. All I/O can be set as one of 4 modes by PxMy register: Quasi-Bi mode (Traditional 8051 mode), Push-Pull mode, Input-Only mode and Open-Drain output mode.

I/O reset status can be set by code option as Quasi-Bi mode or Input-Only mode.

In order to improve EMC capability, every input pin has a Schmitt Trigger. Even enter Power-down mode, Schmitt Trigger is never off.

For SH79F1612B, some I/O pins can share with alternative functions. There exists a priority rule in CPU to avoid these functions be conflict when all the functions are enabled. (Refer to **Port Share** Section for details). Only when the other function is turned off, it allows setting the corresponding register to change the I/O mode.

7.7.2 Register

Table 7.15 Port Control Register

E2H, E4H, E5H EAH, ECH, EDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1M0 (EAH)	P1M07	P1M06	P1M05	P1M04	P1M03	P1M02	P1M01	P1M00
P1M1 (E2H)	P1M17	P1M16	P1M15	P1M14	P1M13	P1M12	P1M11	P1M10
P3M0 (ECH)	P3M07	-	P3M05	P3M04	P3M03	P3M02	P3M01	P3M00
P3M1 (E4H)	P3M17	-	P3M15	P3M14	P3M13	P3M12	P3M11	P3M10
P4M0 (EDH)	-	-	-	-	-	P4M02	P4M01	P4M00
P4M1 (E5H)	-	-	-	-	-	P4M12	P4M11	P4M10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)*	*	*	*	*	*	*	*	*

** I/O reset status can be set by code option as Quasi-Bi mode or Input-Only mode (high impedance).*

I/O Mode

PxM0n	PxM1n	Description
0	0	Quasi-Bi mode
0	1	Push-Pull mode
1	0	Input-Only mode(high impedance)
1	1	Open-Drain output mode

(x = 1, 3 or 4   n = 7, 6, 5, 4, 3, 2, 1 or 0)

Table 7.16 Port Data Register

90H-C0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1 (90H)	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P3 (B0H)	P3.7	-	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
P4 (C0H)	-	-	-	-	-	P4.2	P4.1	P4.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	*	*	*	*	*	*	*	*

** I/O data reset status can be set by code option, if as Quasi-Bi mode I/O data reset value is 0FFH or as Input-Only mode. I/O data reset value is 00H.*

Bit Number	Bit Mnemonic	Description
7-0	**Px.y x = 1-4, y = 0-7**	Port Data Register

Note: *All can be configured as N-channel open drain I/O, but voltage provided for this pin can't exceed V_{DD} + 0.3V.*

7.7.3 Port Structure

Quasi-Bi mode

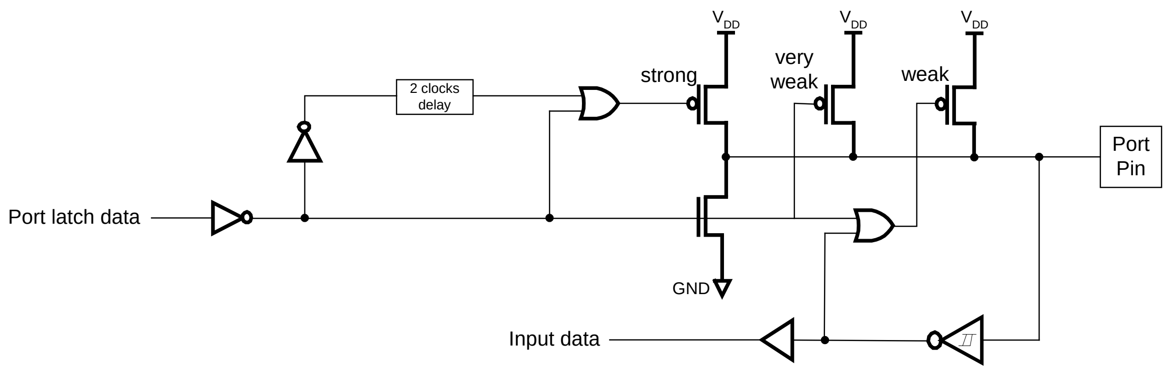
Quasi-Bi I/O has 3 pull-up MOS to adapt to different needs: weak pull-up,very weak pull-up and strong pull-up.

Weak pull-up MOS: When Data register and pin are set 1, this pull-up provides the basic drive current that quasi-bidirectional ports output high. External circuit pull the output-high pin to low, weak pull-up will be off and very weak pull-up will keep on. In order to pull this pin low intensity, external circuit must have sufficient sink current capability to drop the voltage of port below the threshold voltage.

Very weak pull-up MOS: Provide weak pull-up current to pull the pin high when port latch is 1 and the port is floating.

Strong pull-up Mos: When the port latch transition from 0 to 1, strong pull-up is used to speed up the quasi-bi port conversion from logic 0 to logic 1 in almost 2 machine cycles.

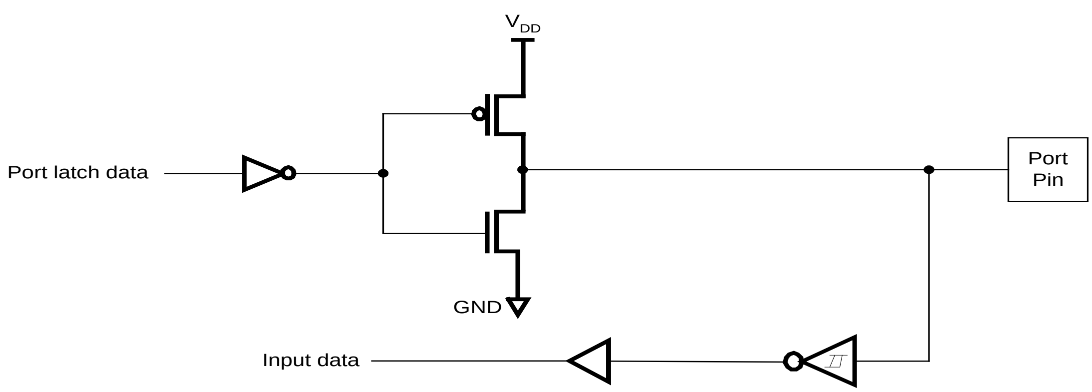
Quasi-bi model port structure diagram is shown below.



Quasi-bi Mode

Push-Pull Mode

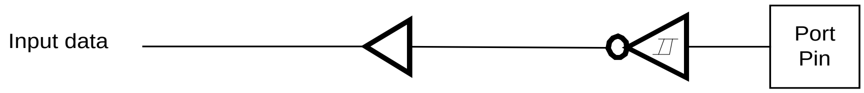
The pull-low structure in push-pull mode is same as open-drain and Quasi-Bi mode, but the port provides a continuous strong pull-up when the port latch is 1. Push-Pull mode port structure diagram is shown below:



Push-Pull Mode

Input-Only Mode

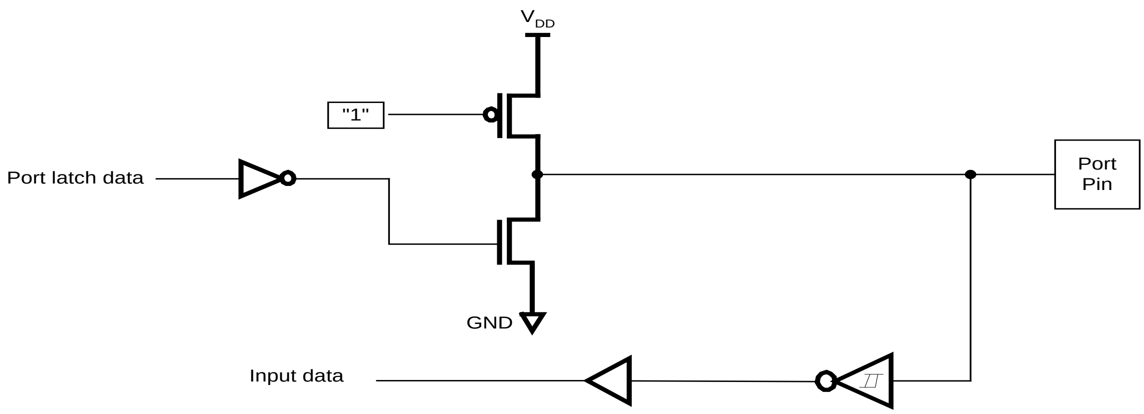
In Input-Only mode port is input only, no output capability. Input-Only mode port structure diagram is shown below:



Input-Only Mode

Open-Drain Mode

In Open-Drain mode the ports have no output high capability. The users should use pull-up resistor to output high. voltage provided for this pin can't exceed V_{DD} + 0.3V. Open-Drain mode port structure diagram is shown below:



Open-Drain Mode

7.7.4 Port Share

The 14/18 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.

When port share function is enabled, any read or write operation to port will only affect the data register while the port pin keeps unchanged until all the share functions are disabled.

PORT1:

- RXD1 (P1.0): EUART1 data input
- TXD1 (P1.1): EUART1 data output
- AN0 - AN7 (P1.0 - P1.7): ADC input channel
- T2 (P1.7): Timer2 external input/baud-rate clock output
- T2EX (P1.6): Timer2 reload/capture control
- VLPD (P1.3): LPD pin
- INT2 (P1.2): external inturrupt 2

Table 7.17 PORT1 Share Table

Pin No.		Priority	Function	Enable bit
SOP/ TSSOP20	SOP16			
14	15	1	AN7	Set ADCH.7 bit in **ADCH** Register and set **ADON** bit in **ADCON** Register, and **SCH[3:0] = 0111**
		2	T2	Set **TR2** bit and **$C/\overline{T2}$** bit in **T2CON** register
		3	P1.7	Above condition is not met
13	14	1	AN6	Set ADCH.6 bit in **ADCH** Register and set **ADON** bit in **ADCON** Register, and **SCH[3:0] = 0110**
		2	T2EX	Set **EXEN2 bit , TR2** bit and **$C/\overline{T2}$** bit in **T2CON** register
		3	P1.6	Above condition is not met
12	13	1	AN5	Set ADCH.5 bit in **ADCH** Register and set **ADON** bit in **ADCON** Register, and **SCH[3:0] = 0101**
		2	P1.5	Clear ADCH.5 bit in **ADCH** Register
11	12	1	AN4	Set ADCH.4 bit in **ADCH** Register and set **ADON** bit in **ADCON** Register, and **SCH[3:0] = 0100**
		2	P1.4	Clear ADCH.4 bit in **ADCH** Register
10	11	1	AN3	Set ADCH.3 bit in **ADCH** Register and set **ADON** bit in **ADCON** Register, and **SCH[3:0] = 0011**
		2	VLPD	Set LPDV bit in LPDCON register
		3	P1.3	Above condition is not met
9	10	1	AN2	Set ADCH.2 bit in **ADCH** Register and set **ADON** bit in **ADCON** Register, and **SCH[3:0] = 0010**
		2	INT2	Set EX2 bit in IEN1 Register
		3	P1.2	Above condition is not met
8	-	1	TXD1	Write operation to SBUF1 register
		2	AN1	Set ADCH.1 bit in **ADCH** Register and set **ADON** bit in **ADCON** Register, and **SCH[3:0] = 0001**
		3	P1.1	Clear ADCH.1 bit in **ADCH** Register and do not write operation to SBUF1 register
7	-	1	RXD1	Set REN1 bit in SCON1
		2	AN0	Set ADCH.0 bit in **ADCH** Register and set **ADON** bit in **ADCON** Register, and **SCH[3:0] = 0000**
		3	P1.0	Clear ADCH.0 bit in **ADCH** Register and clear REN1 bit in SCON1

PORT3:
- RXD (P3.0): EUART data input
- TXD (P3.1): EUART data output
- INT0 (P3.2): external inturrupt 0
- INT1 (P3.3): external inturrupt 1
- T0 (P3.4): Timer0 external input (SOP/TSSOP20)
- T1 (P3.5): Timer1 external input (SOP/TSSOP20)
- PWM (P3.5): PWM output (SOP/TSSOP20)
- T1 (P3.7): Timer 1 external input (SOP16)
- PWM (P3.7): PWM output (SOP16)

Table 7.18 PORT3 Share Table

Pin No.		Priority	Function	Enable bit
SOP/ TSSOP20	SOP16			
17	2	1	RXD	Set REN bit in SCON Register
		2	P3.0	Clear REN bit in SCON Register
18	3	1	TXD	Write to **SBUF** Register
		2	P3.1	Above condition is not met
1	6	1	INT0	Set EX0 bit in IEN0 Register
		2	P3.2	Clear EX0 bit in IEN0 Register
2	7	1	INT1	Set EX1 bit in IEN0 Register
		2	P3.3	Clear EX1 bit in IEN0 Register
3	-	1	T0	Set **TR0** bit in **TCON** Register and Set **C/$\overline{\text{T0}}$** bit in **TMOD** Register
		2	P3.4	Clear **TR0** bit in **TCON** Register and Set **C/$\overline{\text{T0}}$** bit in **TMOD** Register
4	-	1	PWM	Set **PWMSS** and **PWMEN** in **PWMCON** register
		2	T1	Set **TR1** bit in **TCON** Register and Set **C/$\overline{\text{T1}}$** bit in **TMOD** Register
		3	P3.5	Above condition is not met
6	9	1	PWM	Set **PWMSS** and **PWMEN** in **PWMCON** register (SOP16)
		2	T1	Set **TR1** bit in **TCON** Register and Set **C/$\overline{\text{T1}}$** bit in **TMOD** Register (SOP16)
		3	P3.7	Above condition is not met

PORT4:
- RESET (P4.0): Reset pin
- XTAL2 (P4.1): XTAL output

-T0(P3.4): Timer 0 external input (SOP16)
- XTAL1 (P4.2): XTAL input

Table 7.19 PORT4 Share Table

Pin No.		Priority	Function	Enable bit
SOP/ TSSOP20	SOP16			
16	1	1	P4.0	Selected by Code Option
		2	RESET	Selected by Code Option
19	4	1	XTAL2	Selected by Code Option
		2	T0	Set **TR0** bit in **TCON** Register and Set **C/$\overline{\text{T0}}$** bit in **TMOD** Register
		3	P4.1	Above condition is not met
20	5	1	P4.2	Selected by Code Option
		2	XTAL1	Selected by Code Option

7.8 Timer

7.8.1 Feature

- The SH79F1612B has three timers (Timer0, 1, 2)
- Timer0 is compatible with the standard 8051
- Timer1 is compatible with the standard 8051
- Timer2 is compatible with the standard 8052 and has up or down counting and programmable clock output function
- Timer0/1 clock source selectable
- Timer0/1/2 clock source prescaler function
- Timer0/1 compare function

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).

TCLKP1 and TCLKP0 registers are used to select the system clock or 12 frequency division.

When applied as a timer, TCLKS1 and TCLKS0 can be configured in the clock source selection register of Timer x (x = 0,1), and 32.768KHz crystal resonator can be selected as the clock source of Timer 0 and Timer 1 respectively. However, TCLKS1 and TCLKS0 can only be set and operated when 32.768KHz crystal resonator is selected as the clock source.

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

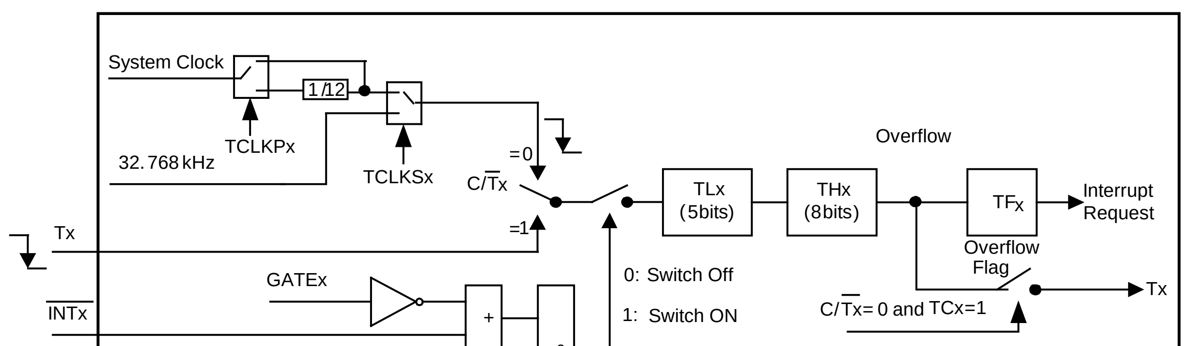
Timer x operate as 13-bit counter/timers in Mode 0. The THx register holds the high eight bits of the 13-bit counter/timer, TLx holds the five low bits TLx.4- TLx.0. The three upper bits (TLx.7 - TLx.5) of TLx are indeterminate and should be ignored when reading. As the 13-bit timer register increments 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.

If C/$\overline{\text{Tx}}$ = 1, high-to-low transitions at the Timer input pin (Tx) will increase the timer/Counter Data register. Else if C/$\overline{\text{Tx}}$ = 0, selects the system clock to increase the timer/Counter Data register. Setting the TRx bit enables the timer when either GATEx = 0, or GATEx = 1 and the input signal $\overline{\text{INTx}}$ is active. Setting GATEx to '1' allows the timer to be controlled by the external input signal $\overline{\text{INTx}}$, facilitating positive pulse width in $\overline{\text{INTx}}$ measurements. Setting TRx does not force the timer to reset. This means that if TRx is set, the timer register will count from the old value that was last stopped by clearing TRx. So the timer registers should be loaded with the desired initial value before the timer is enabled.

When applied as a timer, the TCLKSx (x = 0, 1) bit in the register TCON1 can be configured to select the system clock or 32.768KHz as the clock source of the Timer x (x = 0, 1). And It is only valid when the 32.768KHz crystal resonator is selected in the code option.

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.

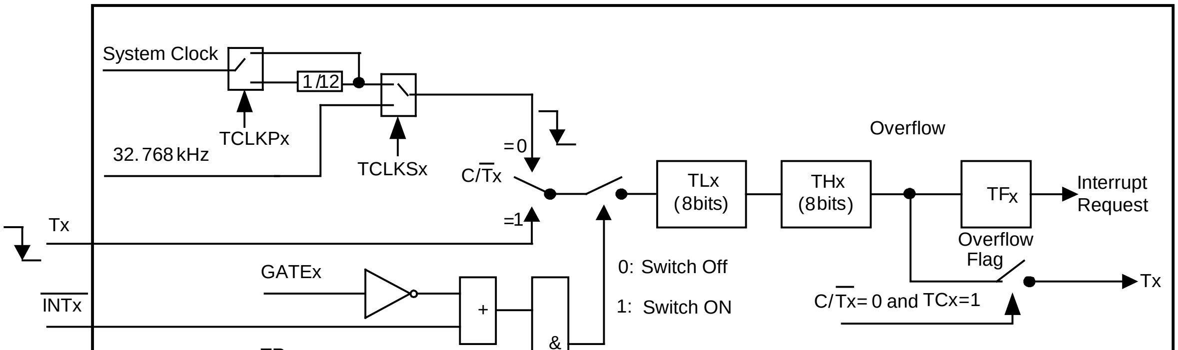
When as Timer, the T0/T1 pin can automatically toggle upon Timer0/1 overflow by configuring TC0/1 in TCON1 Register. The T0/T1 pin is automatically set as output by hardware when TC0/1 is set.



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

Mode1: 16-bit Counter/Timer

Mode1 operation is the same as Mode0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode1 in the same manner as for Mode 0.



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

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

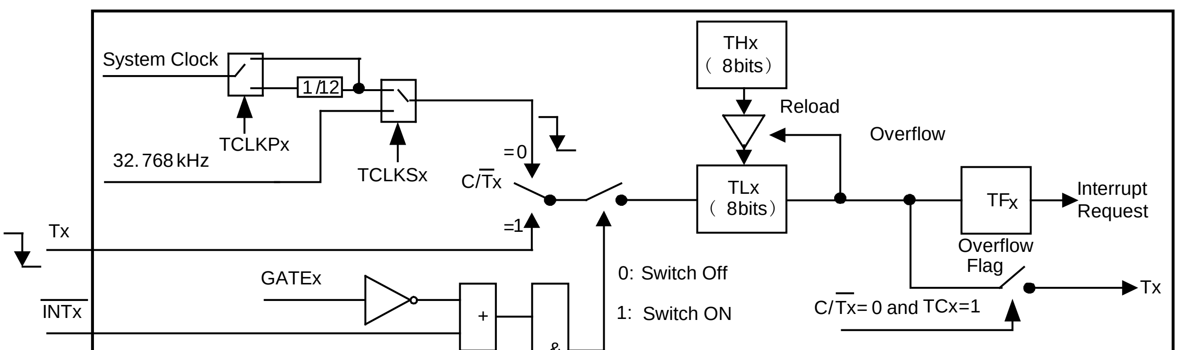
Mode2 configures Timer0 and Timer1 to operate as 8-bit counter/timers with automatic reload of the start value. TLx holds the count and THx holds the reload value. When the counter in TLx overflows from 0xFF to THx, the timer overflow flag TFx is set and the counter in TLx is reloaded from THx. If Timer 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 Mode2 is the same as in Mode0 & Mode1.

When applied as a timer, the TCLKSx (x = 0,1) bit in the register TCON1 can be configured to select the system clock or 32.768KHz as the clock source of the Timer x (x = 0,1). And It is is only valid when the 32.768KHz crystal resonator is selected in the code option.

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.

When as Timer, the T0/T1 pin can automatically toggle upon Timer0/1 overflow by configuring TC0/1 in TCON1 Register. The T0/T1 pin is automatically set as output by hardware when TC0/1 is set.



The Block Diagram of mode 2 of Timerx (x=0, 1)

Mode3: Two 8-bit Counter/Timers (Timer0 Only)

In Mode3, Timer0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. TL0 is controlled using the Timer0 control/status bits in TCON and TMOD: TR0, C/$\overline{T0}$, GATE0 and TF0. TL0 can use either the system clock or 32.768kHz or an external input signal as its time base.

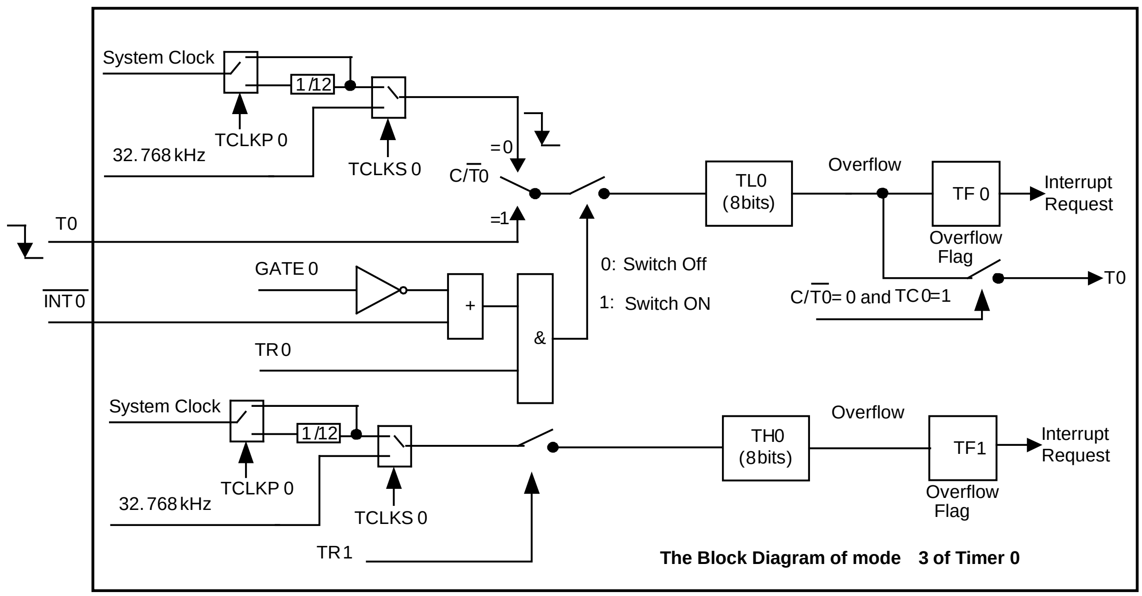
The TH0 is restricted to a timer function sourced by the system clock. TH0 is enabled using the Timer 1 control bit TR1. THx sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

When Timer0 is operating in Mode3, Timer1 can be operated in Modes0, 1 or 2, but it cannot set the TF1 flag and generate an interrupt. The Timer1 overflow can generate baud-rate~~s~~ for the EUART. The TH1 and TL1 register is restricted to a timer function sourced by the system clock, and gate1 is invalid. And the pull high resistor of T1 input pin is also disabled. Timer1 run control is handled through its mode settings, because TR1 is used by Timer0. When the Timer1 is in Mode0, 1, or 2, Timer1 is enable. When the Timer1 is in Mode3, Timer1 is disable.

When applied as a timer, the TCLKS0 bit in the register TCON1 can be configured to select the system clock or 32.768KHz as the clock source of the Timer0. And It is only valid when the 32.768KHz crystal resonator is selected in the code option.

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

When as Timer, the T0 pin can automatically toggle upon Timer0 overflow by configuring TC0 in TCON1 Register. The T0 pin is automatically set as output by hardware when TC0 is set.



The Block Diagram of mode 3 of Timer 0

Note: *While Timer1 is used as baud rate generator, reading or writing TH1/TL1 will affect the accuracy of baud rate, thus might make cause communication error.*

Register

Table 7.20 Timer/Counter x Control Register (x = 0, 1)

88H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
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, 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
3, 1	**IEx x = 0, 1**	**External interrupt x request flag**
2, 0	**ITx x = 0, 1**	**External interrupt x trigger mode select bits**

Table 7.21 Timer/Counter x Mode Register (x = 0, 1)

89H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMOD	GATE1	C/$\overline{T1}$	M11	M10	GATE0	C/$\overline{T0}$	M01	M00
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	**GATEx x = 0, 1**	**Timer x Gate Control bits** 0: Timer x is enabled whenever TRx control bit is set 1: Timer x is enabled only while INTx pin is high and TRx control bit is set
6, 2	**C/$\overline{Tx}$ x = 0, 1**	**Timer x Timer/Counter mode selected bits** 0: Timer Mode, T0 or T1 pin is used as I/O port 1: Counter Mode
5-4 1-0	**Mx[1:0] x = 0, 1**	**Timer x Timer mode selected bits** 00: Mode0, 13-bit up counter/timer, bit7- 5 of TLx is ignored 01: Mode1, 16-bit up counter/timer 10: Mode2, 8-bit auto-reload up counter/timer 11: Mode3 (only for Timer0), two 8-bit up timer

Table 7.22 Timer/Counter x Data 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 (8CH)	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
TL1 (8BH)	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 7.23 Timer/Counter x Control Register1 (x = 0, 1)

CEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON1	-	TCLKS1	TCLKS0	-	TCLKP1	TCLKP0	TC1	TC0
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
6-5	**TCLKSx x = 0, 1**	**Timer x Clock Source Control bits** 0: Select system clock as Timer x Clock Source 1: Select 32.768kHz as Timer x Clock Source
3-2	**TCLKPx x = 0, 1**	**Timer x Clock Source Prescaler bits** 0: Select 1/12 of system clock as Timerx Clock Source 1: Select system clock as Timer x Clock Source
1-0	**TCx x = 0, 1**	**Compare function Enable bits** 0: Disable compare function of Timer x 1: Enable compare function of Timer x

7.8.3 Timer2

The Timer 2 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)

C/$\overline{\text{T2}}$ selects system clock (timer operation) or external pin T2 (counter operation) as the timer clock input. Setting TR2 allows Timer 2/Counter 2 Data Register to increment by the selected input.

Timer2 Mode

Timer 2 has 4 operating modes: Capture/Reload, Auto-reload mode with up or down counter, Baud Rate Generator and Programmable clock-output. These modes are selected by the combination of RCLK, TCLK and CP/RL2.

Timer2 Mode select

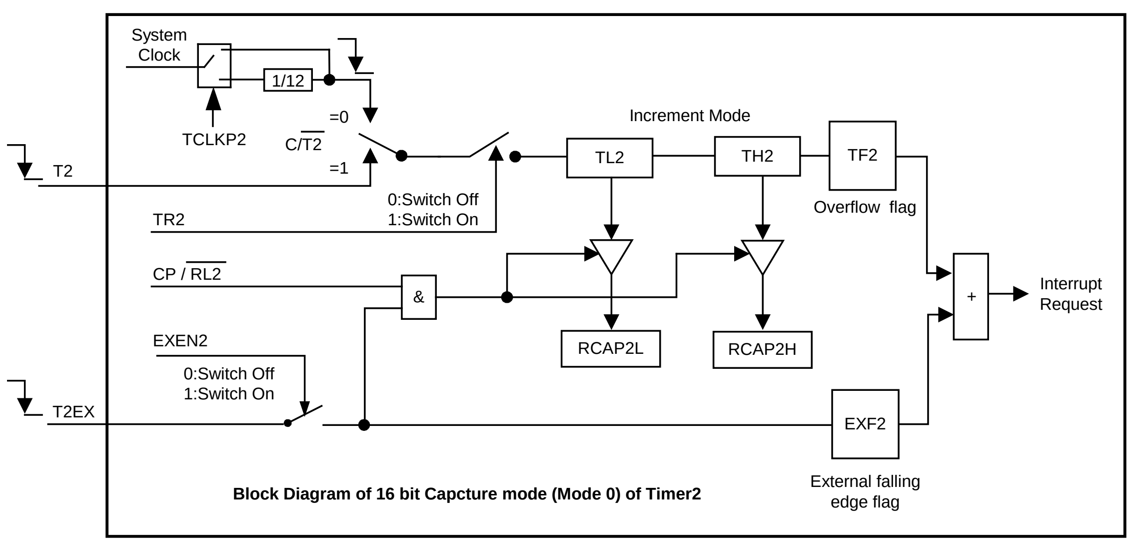
C/$\overline{\text{T2}}$	T2OE	DCEN	TR2	CP/RL2	RCLK	TCLK	Mode	
X	0	X	1	1	0	0	0	16 bit capture
X	0	0	1	0	0	0	1	16 bit auto-reload timer
X	0	1	1	0	0	0		
X	0	X	1	X	1	X	2	Baud-Rate generator
					X	1		
0	1	X	1	X	0	0	3	Programmable clock-output only
					1	X	3	Programmable clock-output, with Baud-rate generator
					X	1		
X	X	X	0	X	X	X	X	Timer2 stop, the T2EX path still enable

Mode0: 16 bit Capture

In the capture mode, two options are selected by bit EXEN2 in T2CON.

If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which will set TF2 on overflow to generate an interrupt if ET2 is enabled.

If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L respectively, In addition, a 1-to-0 transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can also generate an interrupt if ET2 is enabled.



Block Diagram of 16 bit Capcture mode (Mode 0) of Timer2

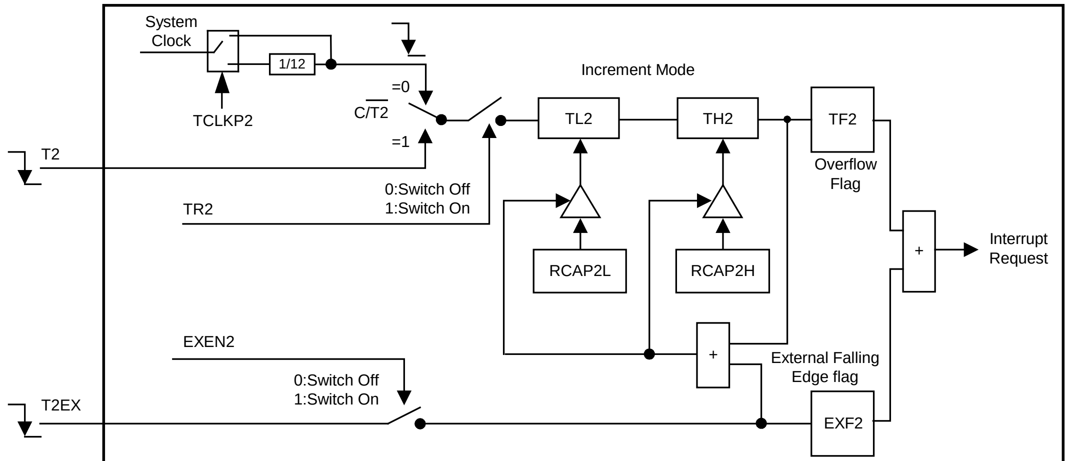
Mode1: 16 bit Auto-reload Timer

Timer2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit in T2MOD. After reset, the DCEN bit is set to 0 so that Timer 2 will default to count up. When DCEN is set, Timer2 can count up or down, depending on the value of the T2EX pin.

When DCEN = 0, two options are selected by bit EXEN2 in T2CON.

If EXEN2 = 0, Timer2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L, which are pressed by software.

If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if ET2 is enabled.



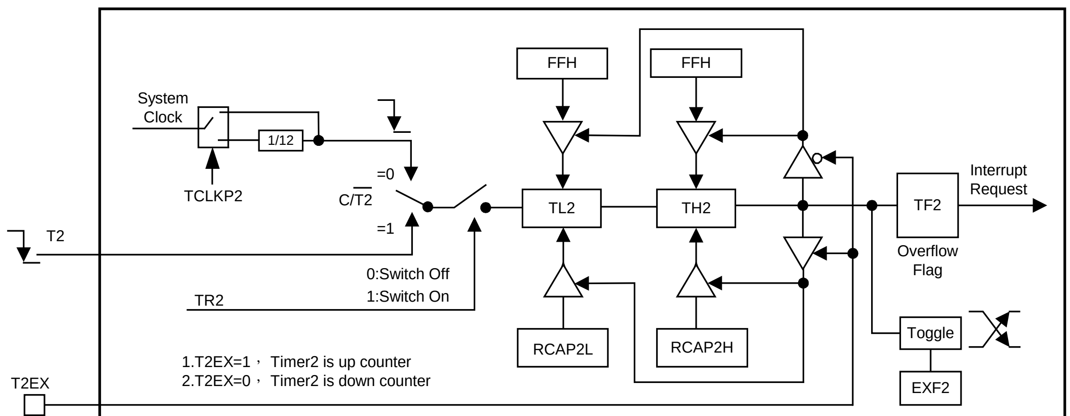
The Block Diagram of Auto Relode Mode (Mode 1)of Timer2 (DCEN=0)

Setting the DCEN bit enables Timer2 to count up or down. When DCEN = 1, the T2EX pin controls the direction of the count, and EXEN2's control is invalid.

A logical "1" at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logical "0" at T2EX makes Timer2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.



The Block Diagram of Auto-Reload Mode (Mode 1) of Timer2 (DCEN=1)

Mode2: Baud-Rate Generator

Timer2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON. The baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer1 is used for the other.

Setting RCLK and/or TCLK will put Timer2 into its baud rate generator mode, which is similar to the auto-reload mode.

Over flow of Timer2 will causes the Timer2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L that preset by software. But this will not generate an interrupt.

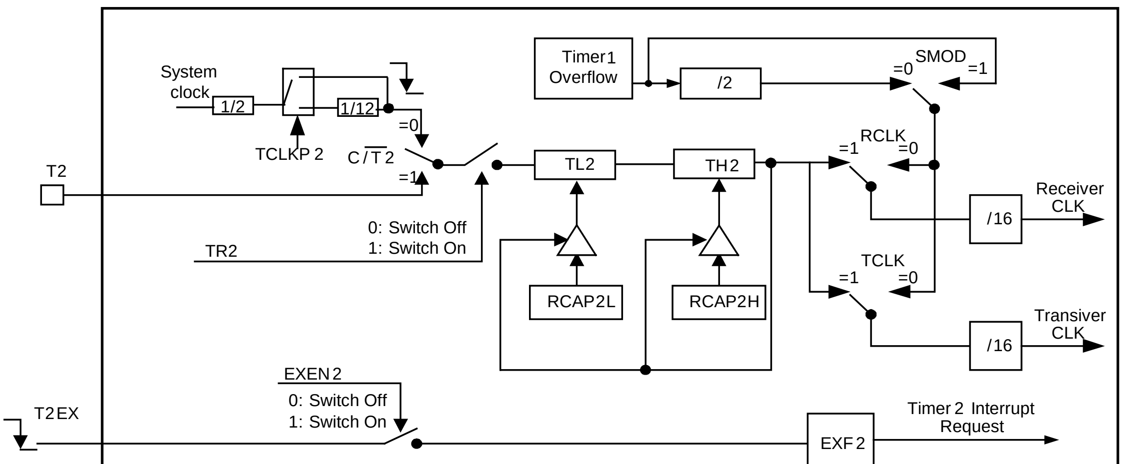
If EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload. Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

The baud rates in EUART Mode1 and 3 are determined by Timer2's overflow rate according to the following equation.

$$BaudRate = \frac{1}{2\times 12\times 16}\times\frac{f_{SYS}}{65536-[RCAP2H, RCAP2L]}; \quad C/\overline{T2} = 0, TCLKP2 = 0$$

$$BaudRate = \frac{1}{2\times 16}\times\frac{f_{SYS}}{65536-[RCAP2H, RCAP2L]}; \quad C/\overline{T2} = 0, TCLKP2 = 1$$

$$BaudRate = \frac{1}{16}\times\frac{f_{T2}}{65536-[RCAP2H, RCAP2L]}; \quad C/\overline{T2} = 1$$



The Block Diagram of Baund-Rate Generator (Mode2) of Timer2

Mode3: Programmable Clock Output

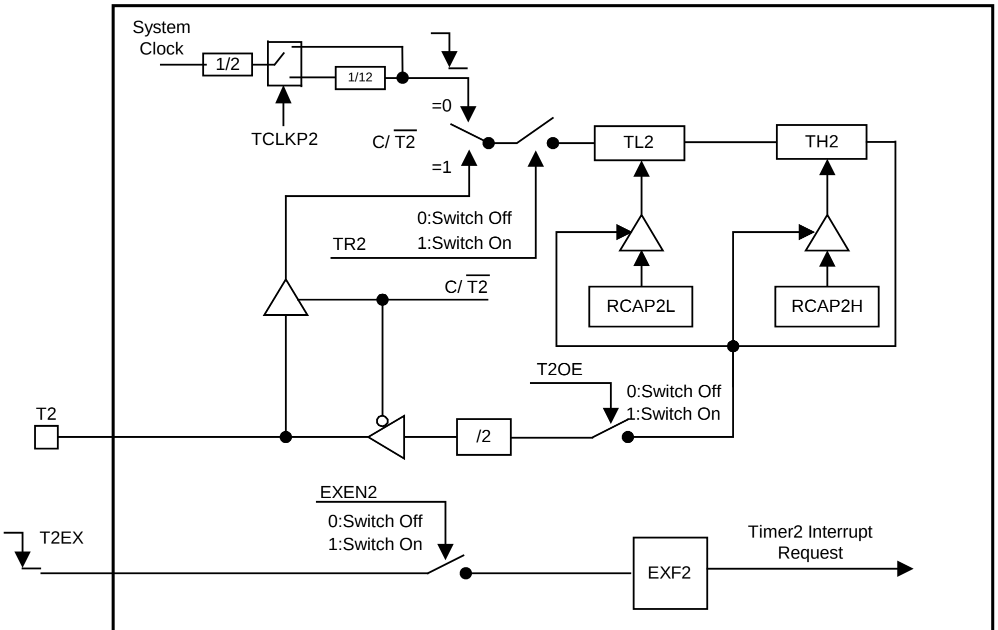
A 50% duty cycle clock can be programmed to come out on P1.7. To configure the Timer 2 as a clock generator, bit C/$\overline{T2}$ must be cleared and bit T2OE must be set. Bit TR2 starts and stops the timer.

In this mode T2 will output a 50% duty cycle clock,

$$Clock\ \ Out\ \ Frequency = \frac{1}{2\times 2\times 12} \times \frac{f_{SYS}}{65536-[RCAP2H, RCAP2L]} \quad ;\ \text{TCLKP2 = 0}$$

$$Clock\ \ Out\ \ Frequency = \frac{1}{2\times 2} \times \frac{f_{SYS}}{65536-[RCAP2H, RCAP2L]} \quad ;\ \text{TCLKP2 = 1}$$

Timer2 overflow will not generate an interrupt, so it is possible to use Timer2 as a baud-rate generator and a clock output simultaneously with the same frequency.



The Block Diagram of Programmable Clock output (Mode 3) of Timer2

Note:

(1) Both TF2 and EXF2 can cause timer2 interrupt request, and they have the same vector address.

(2) TF2 and EXF2 are 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 & EXF2 to 0.

(3) When EA = 1 & ET2 = 1, setting TF2 or EXF2 as 1 will cause a timer2 interrupt.

(4) While Timer2 is used as baud rate generator, writing TH2/TL2, writing RCAPH2/RCAPL2 will affect the accuracy of baud rate, thus might make cause communication error.

Register

Table 7.24 Timer2 Control Register

C8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	$CP/\overline{RL2}$
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	**TF2**	**Timer2 overflow flag bit** 0: No overflow 1: Overflow (Set by hardware if RCLK = 0 & TCLK = 0)
6	**EXF2**	**External event input (falling edge) from T2EX pin detected flag bit** 0: No external event input (Must be cleared by software) 1: Detected external event input (Set by hardware if EXEN2 = 1)
5	**RCLK**	**EUART0 Receive Clock control bit** 0: Timer 1 generates receiveing baud-rate 1: Timer 2 generates receiveing baud-rate
4	**TCLK**	**EUART0 Transmit Clock control bit** 0: Timer1 generates transmitting baud-rate 1: Timer 2 generates transmitting baud-rate
3	**EXEN2**	**External event input (falling edge) from T2EX pin used as Reload/Capture trigger enable/disable control bit** 0: Ignore events on T2EX pin 1: Cause a capture or reload when a negative edge on T2EX pin is detected, when Timer2 is not used to clock the EUART (T2EX always has a pull up resistor)
2	**TR2**	**Timer2 start/stop control bit** 0: Stop Timer2 1: Start Timer2
1	**$C/\overline{T2}$**	**Timer2 Timer/Counter mode selected bit** 0: Timer Mode, T2 pin is used as I/O port 1: Counter Mode, the internal pull-up resister is turned on
0	**$CP/\overline{RL2}$**	**Capture/Reload mode selected bit** 0: 16 bits timer/counter with reload function 1: 16 bits timer/counter with capture function

Table 7.25 Timer2 Mode Control Register

C9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2MOD	TCLKP2	-	-	-	-	-	T2OE	DCEN
R/W	R/W	-	-	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	-	-	-	-	-	0	0

Bit Number	Bit Mnemonic	Description
7	**TCLKP2**	**Timer2 Clock Source Prescaler bits** 0: Select 1/12 of system clock as Timer2 Clock Source 1: Select system clock as Timer2 Clock Source
1	**T2OE**	**Timer2 Output Enable bit** 0: Set P1.7/T2 as clock input or I/O port 1: Set P1.7/T2 as clock output (Baud-Rate generator mode)
0	**DCEN**	**Down Counter Enable bit** 0: Disable Timer2 as up/down counter, Timer2 is an up counter 1: Enable Timer2 as up/down counter

Table 7.26 Timer2 Reload/Capture & Data Register

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** **RCAP2H.x**	**Timer2 Reload/ Capturer Data, x = 0 - 7**
7-0	**TL2.x** **TH2.x**	**Timer 2 Low & High byte counter, x = 0 - 7**

7.9 Interrupt

7.9.1 Feature

- 13 interrupt sources
- 4 interrupt priority levels

7.9.2 Description

The SH79F1612B provides total 13 interrupt sources: 3 external interrupts (INT0/1/2), 3 timer interrupts (Timer0, 1, 2), LPD interrupt, 2 EUART interrupt, ADC Interrupt, SCM interrupt, and PWM interrupts.

7.9.3 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.

Table 7.27 Primary Interrupt Enable Register

A8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EA	EADC	ET2	ES0	ET1	EX1	ET0	EX0
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	EA	**All interrupt enable bit** 0: Disable all interrupt 1: Enable all interrupt
6	EADC	**ADC interrupt enable bit** 0: Disable ADC interrupt 1: Enable ADC interrupt
5	ET2	**Timer2 overflow interrupt enable bit** 0: Disable timer2 overflow interrupt 1: Enable timer2 overflow interrupt
4	ES0	**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	EX1	**External interrupt 1 enable bit** 0: Disable external interrupt1 1: Enable external interrupt1
1	ET0	**Timer0 overflow interrupt enable bit** 0: Disable Timer0 overflow interrupt 1: Enable Timer0 overflow interrupt
0	EX0	**External interrupt 0 enable bit** 0: Disable external interrupt0 1: Enable external interrupt0

Table 7.28 Secondary Interrupt Enable Register

A9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN1	ELPD	-	EPWM	ESCM	ES1	EX2	-	-
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
7	**ELPD**	**LPD interrupt enable bit** 0: Disable LPD interrupt 1: Enable LPD interrupt
5	**EPWM**	**PWM interrupt enable bit** 0: Disable PWM interrupt 1: Enable PWM interrupt
4	**ESCM**	**SCM interrupt enable bit** 0: Disable SCM interrupt 1: Enable SCM interrupt
3	**ES1**	**EUART1 interrupt enable bit (SOP16 Pin no)** 0: Disable EUART1 interrupt 1: Enable EUART1 interrupt
2	**EX2**	**Enternal interrupt2 enable bit** 0: Disable external interrupt2 1: Enable external interrupt2

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 Table bellow.

For **external interrupt (INT0/1/2)**, when an external interrupt0/1/2 is generated, if the interrupt was edge trigged, the flag (IE0-2 in TCON) 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 and bit EXF2 in T2CON register, which is set by hardware. None of these flags can be cleared by hardware when the service routine is vectored. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, so the flag must be cleared by software.

The **EUART/EUART1 interrupt** is generated by the logical OR of flag RI/RI1 and TI/TI1 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 SCM register, which is set by hardware. And the flag can only be cleared by hardware.

The **ADC interrupt** is generated by ADCIF bit in ADCON. If an interrupt is generated, the converted result in ADCDH/ADCDL will be valid. If continuous compare function in ADC module is Enable, ADCIF will not be set at each conversion, but set if converted result is larger than compare value. The flag must be cleared by software.

The **PWM interrupts** are generated by PWMIF. The flags can be cleared by software.

The **LPD interrupts** are generated by LPDF. The flags is set by hardware, cleared by software.

Table 7.29 Enternal Interrupt Flag Register

88H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
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
7,5	**TFx (x = 0, 1)**	**Timer x overflow flag bit** 0: No overflow 1: Overflow
6,4	**TRx (x = 0, 1)**	**Timer x start/stop control bit** 0: Stop Timer x 1: Start Timer x
3, 1	**IEx (x = 0, 1)**	**External interrupt x request flag bit** 0: No interrupt pending 1: Interrupt is pending
2, 0	**ITx (x = 0, 1)**	**External interrupt x trigger mode selection bit** 0: Low level trigger 1: Falling edge trigger

Table 7.30 External Interrupt Flag Register

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

Bit Number	Bit Mnemonic	Description
3-2	**IT2[1:0]**	**External interrupt 2 trigger mode selection bit** 00: Low Level trigger 01: Trigger on falling edge 10: Trigger on rising edge 11: Trigger on both edge
0	**IE2**	**External interrupt 2 request flag bit** 0: No interrupt pending 1: Interrupt is pending

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. But the OVL NMI interrupt has the highest Priority Level (except RESET) of all the interrupt sources, with no IPH/IPL control. 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 requests of the same priority level are pending at the start of an instruction cycle, 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 7.31 Interrupt Priority Control Register

B8H, B4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IPL0	-	PADCL	PT2L	PS0L	PT1L	PX1L	PT0L	PX0L
IPH0	-	PADCH	PT2H	PS0H	PT1H	PX1H	PT0H	PX0H
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
B9H, B5H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
IPL1	PLPDL	-	PPWML	PSCML	PSL1	PX2L	-	-
IPH1	PLPDH	-	PPWMH	PSCMH	PSH1	PX2H	-	-
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
7-0	**PxxxL/H**	Corresponding interrupt source xxx's priority level selection bits

7.9.7 Interrupt Handling

The interrupt flags are sampled and polled at the fetch cycle of 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, provided this hardware-generated LCALL 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. This ensures that the instruction in progress is completed before vectoring to any service routine.

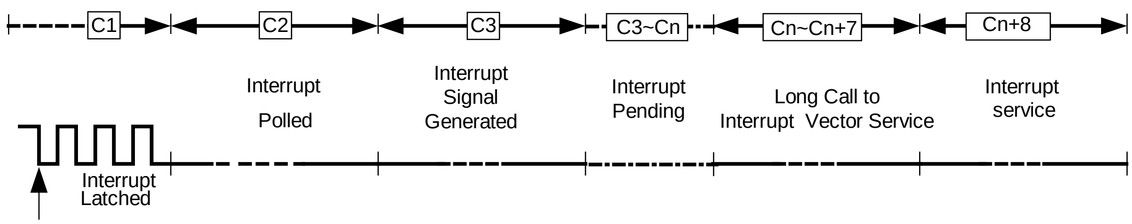
The instruction in progress is RETI. This ensures that if the instruction in progress is RETI then at least one more instruction except RETI will be executed before any interrupt is vectored to; this delay guarantees that the CPU can observe the changes of the interrupt status.

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 Timing

The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the program counter with corresponding address that depends on the source of the interrupt being vectored too, as shown in Interrupt Summary table.

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. Note that the RETI instruction is very important because it informs the processor that the program left the current interrupt service. A simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt with this priority was still in progress. In this case, no interrupt of the same or lower priority level would be acknowledged.

7.9.8 Interrupt Response Time

If an interrupt is recognized, its request flag is set in every machine cycle after recognize. The value will be polled by the 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 LCALL to the requested service routine will be the next instruction to be executed. Else the interrupt will pending. The call itself takes 7 machine cycles. Thus a minimum of 3+7 complete machine cycles will elapse between activation and external interrupt request and the beginning of execution of the first instruction of the service routine.

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 nature 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

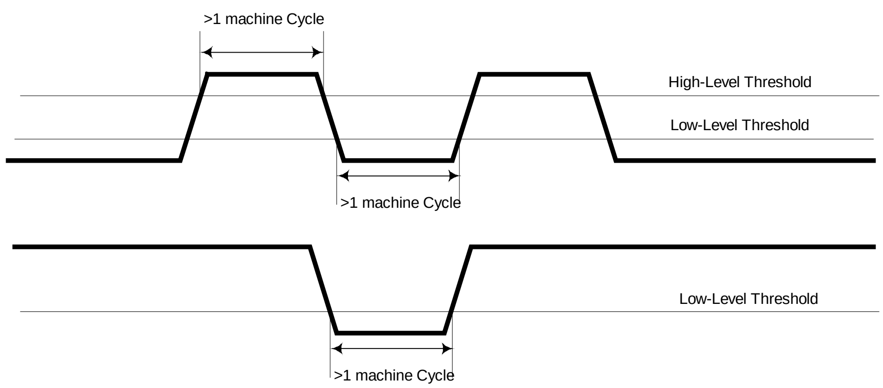
The SH79F1612B has 3 external interrupt inputs. External interrupt0-2 each has one vector address. These external interrupts can be programmed to be level-triggered or edge-triggered by clearing or setting bit IT1 or IT0 in register TCON and register EXF1. If ITx = 0 (x = 0, 1), external interrupt INTx (x = 0, 1) is triggered by a low level detected. If ITx = 1 (x = 0, 1), external interrupt INTx (x = 0, 1) is edge triggered. In this mode if consecutive samples of the INTx (x = 0, 1) pin show a high level in one cycle and a low level in the next cycle, interrupt request flag in register EXF1 is set, causing an interrupt request. Since the external interrupt pins are sampled once each machine cycle, an input high or low level should be held for at least one machine cycle to ensure proper sampling.

If the external interrupt is edge-triggered, the external source has to hold the request pin high for at least one machine cycle, and then hold it low for at least one machine cycle. This is to ensure that the transition is detected and that interrupt request flag is set. Notice that IE0-1 is automatically cleared by CPU when the service routine is called.

If the external interrupt is level-triggered, the external source must hold the request active until the requested interrupt is generated, which will take 2 machine cycles. If the external interrupt is still asserted when the interrupt service routine is completed, another interrupt will be generated. It is not necessary to clear the interrupt flag IEx (x = 0, 1, 2) when the interrupt is level sensitive, it simply tracks the input pin level.

When SH79F1612B 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.

Note: *IE0-2 is automatically cleared by CPU when the service routine is called.*



7.9.10 Interrupt Summary

Source	Vector Address	Enable bits	Flag bits	Polling Priority	Interrupt number (c language)
Reset	0000H	-	-	0 (highest)	-
INT0	0003H	EX0	IE0	1	0
Timer0	000BH	ET0	TF0	2	1
INT1	0013H	EX1	IE1	3	2
Timer1	001BH	ET1	TF1	4	3
EUART	0023H	ES	RI+TI	5	4
Timer2	002BH	ET2	TF2+EXF2	6	5
ADC	0033H	EADC	ADCIF	7	6
INT2	004BH	EX2	IE2	8	9
EUART1	0053H	ES1	RI1+TI1	9	10
SCM	005BH	ESCM	SCMIF	10	11
PWM	0063H	EPWM	PWMIF	11	12
LPD	0073H	ELPD	LPDF	12	14

8. Enhanced Fucntion

8.1 EUART/EUART1

8.1.1 Feature

- The SH79F1612B has two enhanced EUART which are compatible with the conventional 8051 (The SOP16 package contains only one EUART)
- The baud rate can be selected from the divided clock of the system clock, or Timer1/2 overflow rate
- Euart1 has its own baud rate generator. The baud rate can be selected as 1/16 of system clock frequency division or overflow rate of its own baud rate generator
- Enhancements over the standard 8051 the EUART include Framing Error detection and automatic address recognition
- The EUART/EUART1 can be operated in four modes

8.1.2 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. The Timer1/2 should also be initialized if the mode 1 or the mode 3 is used.

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

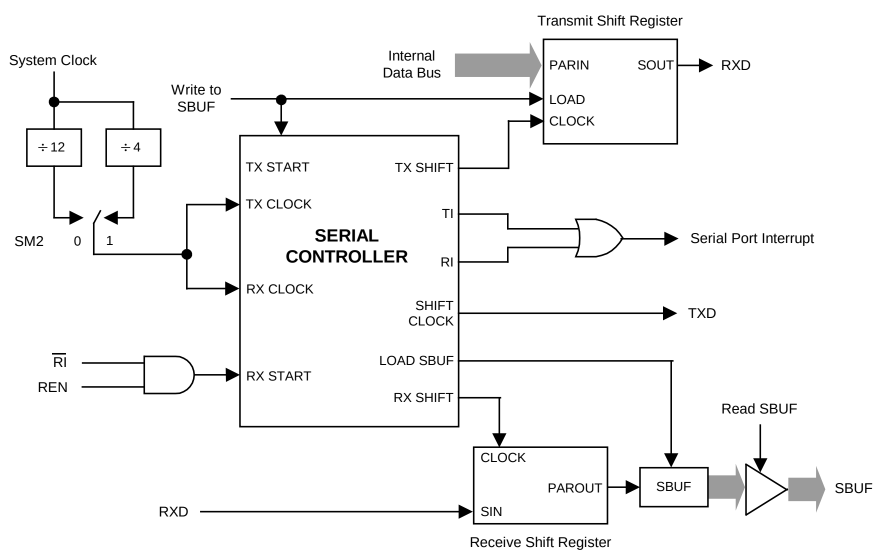
SM0	SM1	Mode	Type	Baud Clock	Frame Size	Start Bit	Stop Bit	9^{th} bit
0	0	0	Sych	f_{SYS}/(4 or 12)	8 bits	NO	NO	None
0	1	1	Ansych	Timer 1 or 2 overflow rate/(16 or 32)	10 bits	1	1	None
1	0	2	Ansych	f_{SYS}/(32 or 64)	11 bits	1	1	0, 1
1	1	3	Ansych	Timer 1 or 2 overflow rate/(16 or 32)	11 bits	1	1	0, 1

Mode0: Synchronous Mode, Half duplex

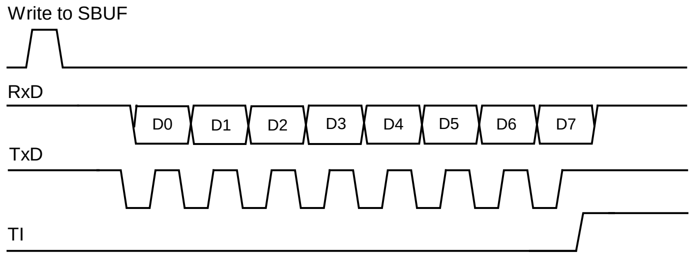
This mode provides synchronous communication with external devices. In this mode serial data is transmitted and received on the RxD line. TxD is used to output the shift clock. The TxD clock is provided by the SH79F1612B whether the device is transmitting or receiving. This mode is therefore a half duplex 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 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. The SHIFT CLOCK is used to shift data into and out of the SH79F1612B.

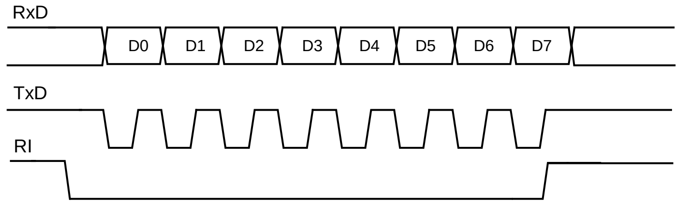


Any instruction that uses SBUF as a destination register ("write to SBUF" signal) will start the transmission. The next system clock tells the Tx control block to commence a transmission. The data shift occurs at the falling edge of the SHIFT CLOCK, and the contents of the transmit shift register is shifted one position 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

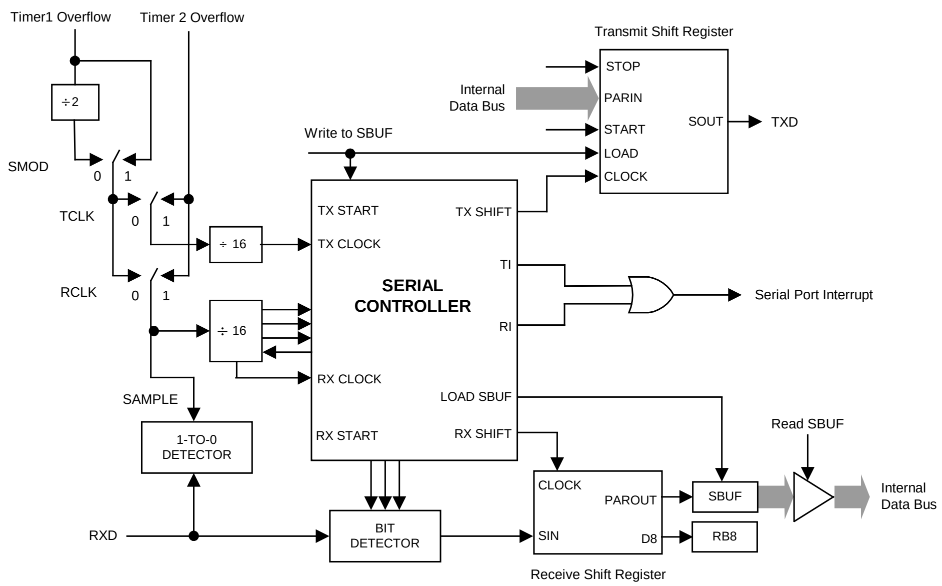
Reception is initiated by the condition REN (SCON.4) = 1 and RI (SCON.0) = 0. The next system clock activates RECEIVE. The data latch occurs at the rising edge of the SHIFT CLOCK, and the contents of the receive shift register are shifted one position to the left. After the receiving of all 8 bits into 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.



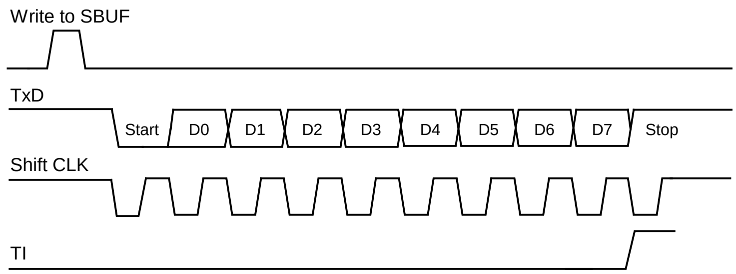
Receive Timing of Mode 0

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 baud rate in this mode is variable. The serial receive and transmit baud rate can be programmed to be 1/16 of the Timer1/2 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

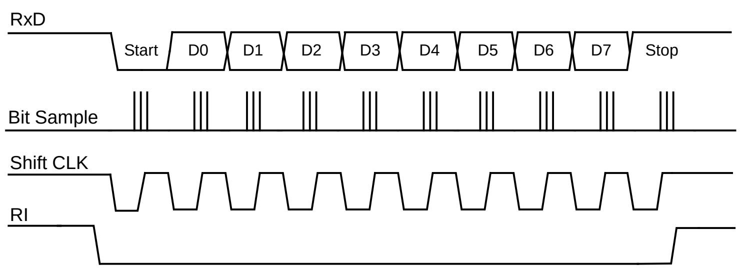
Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. This helps 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 states 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 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 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 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.

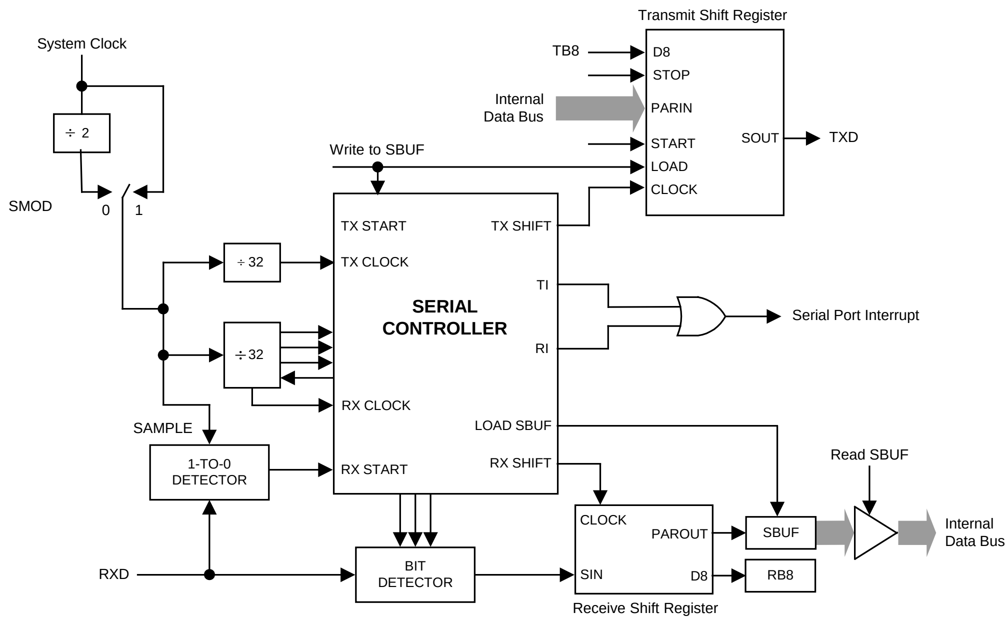
At the time, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



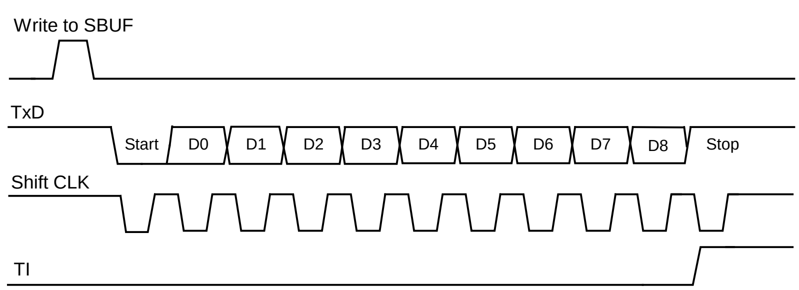
Receive Timing of Mode 1

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). Mode2 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

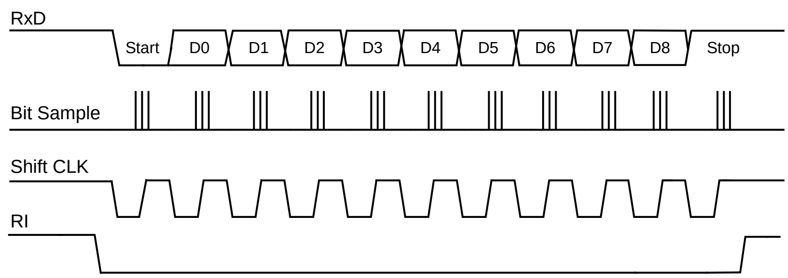
Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. 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 7^{th}, 8^{th} and 9^{th} 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 9^{th} bit = 1 and the received byte accords with Given Address

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

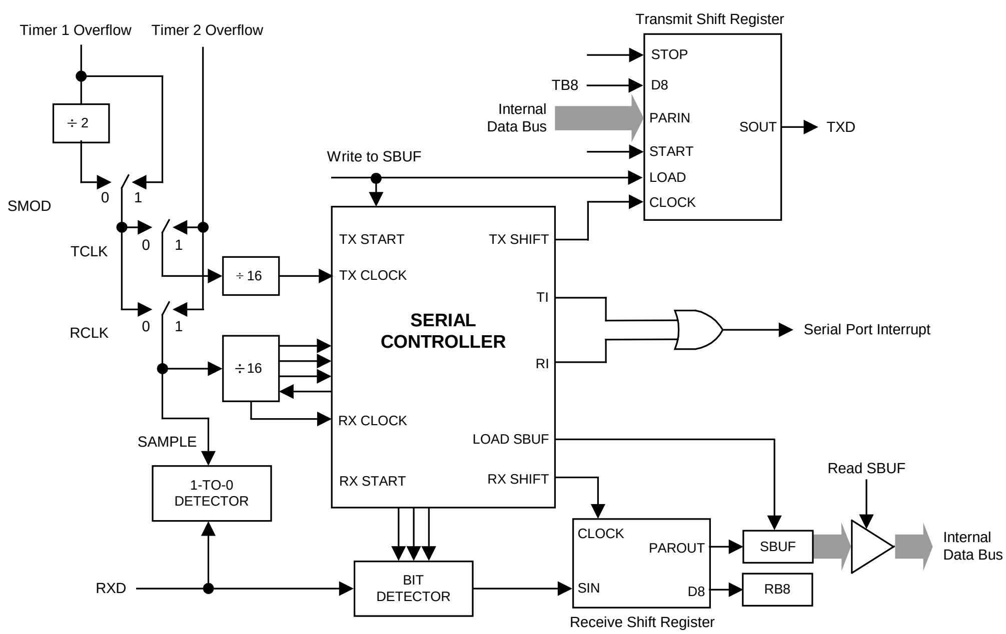
At the time, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



Receive Timing of Mode 2

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

Mode3 uses transmission protocol of the Mode 2 and baud rate generation of the Mode1.



8.1.3 Baud Rate Generate

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.

In Mode1 & Mode3, the baud rate can be selected from Timer1/2 overflow rate.

The Mode1 & 3 baud rate equations are shown below, where [RCAP2H, RCAP2L] is the 16-bit reload register for Timer2, SMOD is the EUART baud rate doubler (PCON.7), T1CLK is the clock source of Timer1. T2CLK is the clock source of Timer2.

$\mathrm{BaudRate} = \frac{2^{SMOD}}{32} \times \frac{f_{T1}}{256 - TH1}$, Baud Rate using Timer1, working in Mode2.

$\mathrm{BaudRate} = \frac{1}{2 \times 16} \times \frac{f_{T2}}{65536 - [\mathrm{RCAP2H, RCAP2L}]}$, Baud Rate using Timer2, the clock source of Timer2 is system clock.

$\mathrm{BaudRate} = \frac{1}{16} \times \frac{f_{T2}}{65536 - [\mathrm{RCAP2H, RCAP2L}]}$, Baud Rate using Timer2, the clock source of Timer2 is input clock of T2 pin

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.

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

8.1.4 Multi-Processor Communication

Software Address Recognition

Modes 2 and 3 of the EUART have a special provision for multi-processor communication. In these modes, 9 data bits are received. The 9th bit goes into RB8. Then a stop bit follows. The EUART can be programmed such that when the stop bit is received, the EUART interrupt will be activated (i.e. the request flag RI is set) only if RB8 = 1. This feature is enabled by setting the bit SM2 in SCON.

A way to use this feature in multiprocessor communications is as follows. If the master processor wants to transmit a block of data to one of the several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte.

With SM2 = 1, no other slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. After having received a complete message, the slave sets SM2 again. The slaves that were not addressed leave their SM2 set and go on with their business, ignoring the incoming data bytes.

Note: *In mode 0, SM2 is used to select baud rate doubling. In mode 1, SM2 can be used to check the validity of the stop bit. If SM2 = 1 in mode 1, the receive interrupt will not be activated unless a valid stop bit is received.*

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 9^{th} 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 9^{th} 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 0 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.

8.1.5 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.

Break Detection

A break is detected when any 11 consecutive bits are sensed low. Since a break condition also satisfies the requirements for a framing error, a break condition will also result in reporting a framing error. Once a break condition has been detected, the EUART will go into an idle state and remain in this idle state until a valid stop bit (rising edge on RxD line) has been received.

8.1.6 Register

Table 8.1 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** If set in mode 1 & 3, the baud-rate of EUART is doubled if using time4 as baud-rate generator If set in mode 2, the baud-rate of EUART is doubled
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**

EUART related SFR

Table 8.2 EUART Control & Status Register

98H	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 mode 0, baud-rate is 1/12 of system clock In mode 1, disable stop bit validation check, any stop bit will set RI to generate interrupt In mode 2 & 3, any byte will set RI to generate interrupt 1: In mode 0, baud-rate is 1/4 of system clock In mode 1, Enable stop bit validation check, only valid stop bit (1) will set RI to generate interrupt In mode 2 & 3, only address byte (9th bit = 1) will set RI to generate interrupt

(to be continued)

(continue)

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 mode 2 & 3 of EUART, set or clear by software**
2	RB8	**The 9th bit to be received in mode 1,2 & 3 of EUART** In mode 0, RB8 is not used In mode 1, if receive interrupt occurs, RB8 is the stop bit that was received In modes 2 & 3 it is the 9^{th} bit that was received
1	TI	**Transmit interrupt flag of EUART** 0: cleared by software 1: Set by hardware at the end of the 8^{th} bit time in mode 0, or at the beginning of the stop bit in other modes
0	RI	**Receive interrupt flag of EUART** 0: cleared by software. 1: Set by hardware at the end of the 8^{th} bit time in mode 0, or during the stop bit time in other modes

Table 8.3 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 8.4 EUART Slave Address & Address Mask Register

9AH-9BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SADDR	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
SADEN	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

8.1.7 EUART1

EUART1 Mode Description

The EUART1 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.

EUART1 Mode Summary

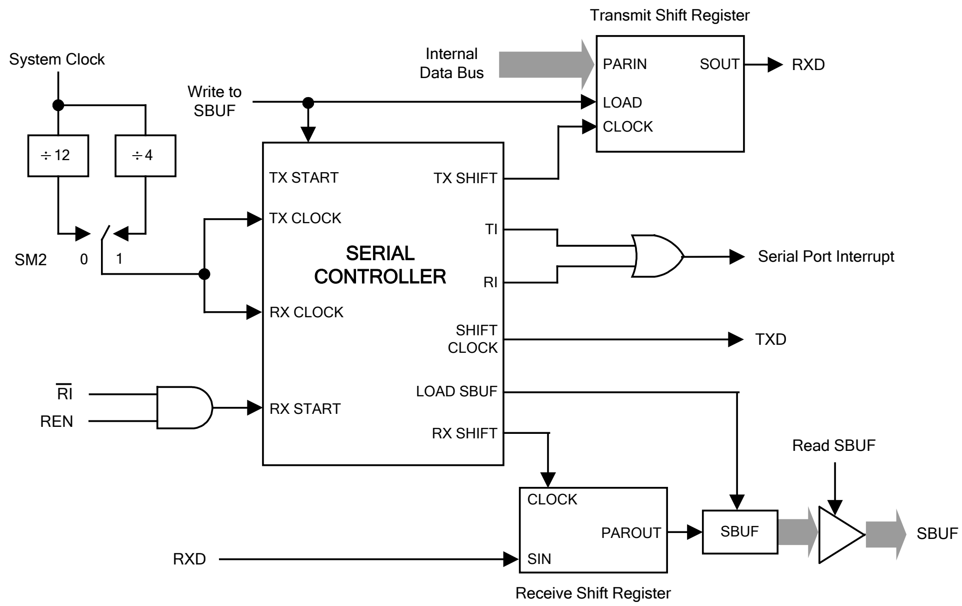
SM0	SM1	Mode	Type	Baud Clock	Frame Size	Start Bit	Stop Bit	9th bit
0	0	0	Sych	f_{SYS}/(4 or 12)	8 bits	NO	NO	None
0	1	1	Ansych	overflow rate of baud rate generator /16	10 bits	1	1	None
1	0	2	Ansych	f_{SYS}/(32 or 64)	11 bits	1	1	0, 1
1	1	3	Ansych	overflow rate of baud rate generator /16	11 bits	1	1	0,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

Mode0: Synchronous Mode, Half duplex

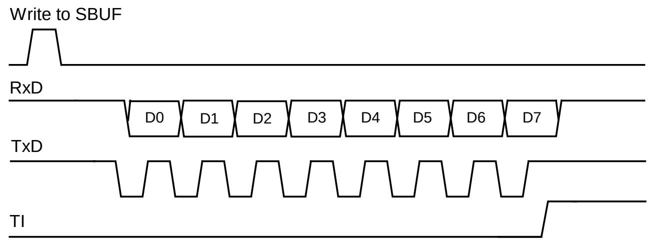
This mode provides synchronous communication with external devices. In this mode serial data is transmitted and received on the RxD line. TxD is used to output the shift clock. The TxD clock is provided by the SH79F1612B whether the device is transmitting or receiving. This mode is therefore a half duplex 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 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. The SHIFT CLOCK is used to shift data into and out of the SH79F1612B.

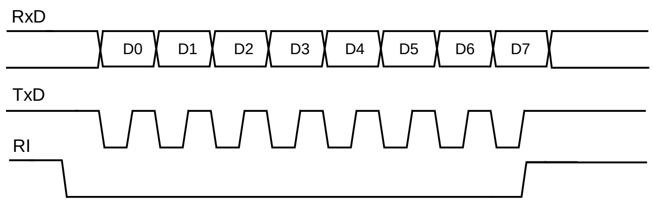


Any instruction that uses SBUF as a destination register ("write to SBUF" signal) will start the transmission. The next system clock tells the Tx control block to commence a transmission. The data shift occurs at the falling edge of the SHIFT CLOCK, and the contents of the transmit shift register is shifted one position 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

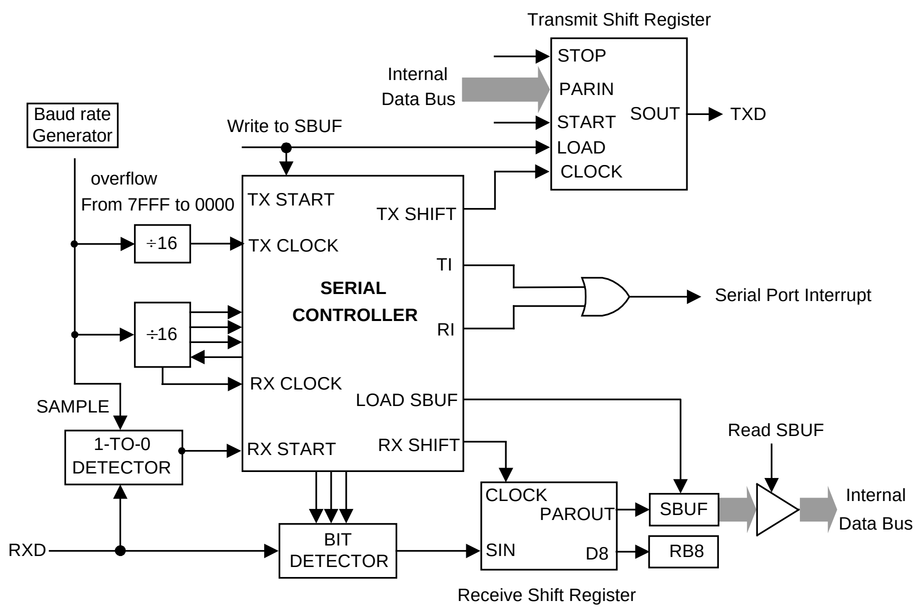
Reception is initiated by the condition REN (SCON.4) = 1 and RI (SCON.0) = 0. The next system clock activates RECEIVE. The data latch occurs at the rising edge of the SHIFT CLOCK, and the contents of the receive shift register are shifted one position to the left. After the receiving of all 8 bits into 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.



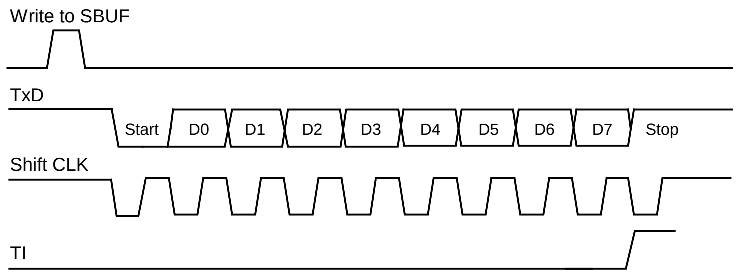
Receive Timing of Mode 0

Mode1: 8-Bit EUART1, 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 baud rate in mode1 is its own baud rate generator overflow rate/16. 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

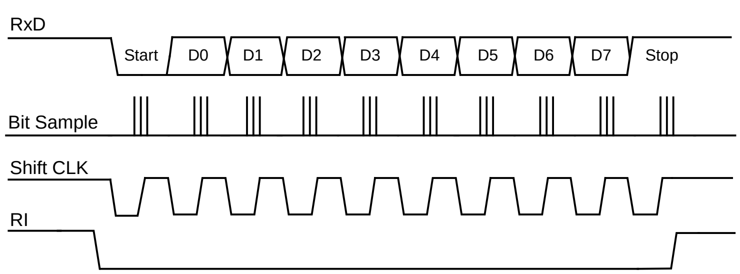
Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. This helps 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 states 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 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 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 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.

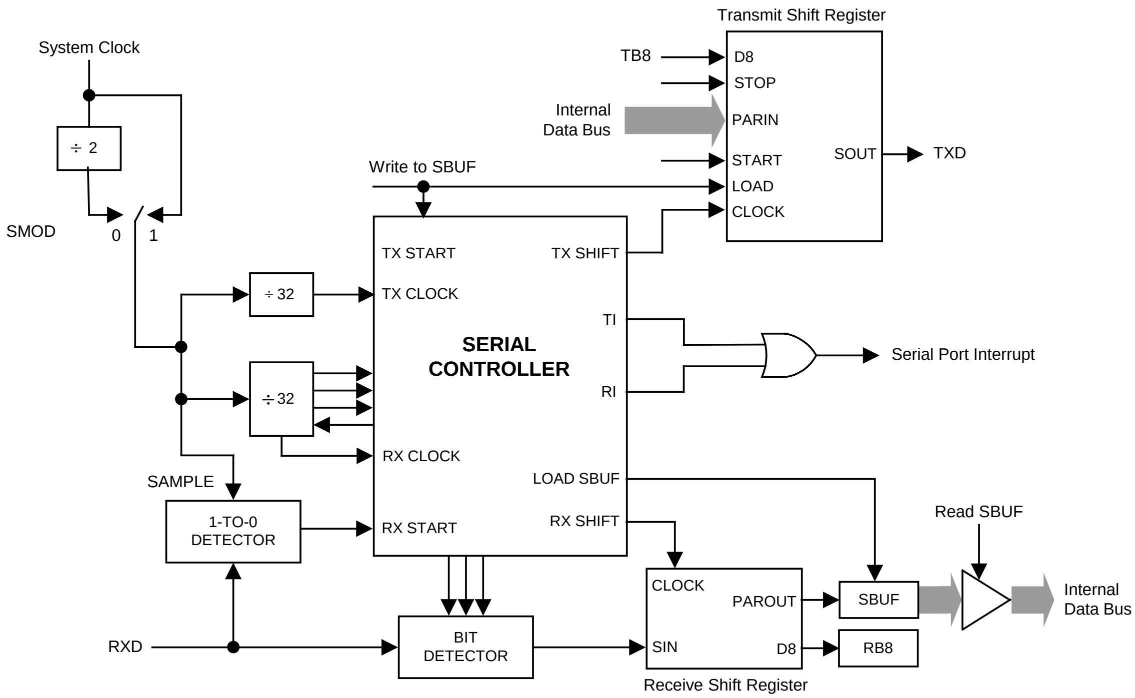
At the time, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



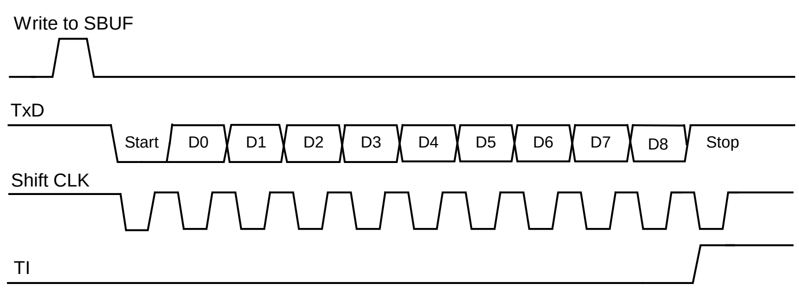
Receive Timing of Mode 1

Mode2: 9-Bit EUART1, 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 9^{th} 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 9^{th} 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 9^{th} 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 9^{th} 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 11^{th} rollover of the divide-by-16 counter after a write to SBUF.



Send Timing of Mode 2

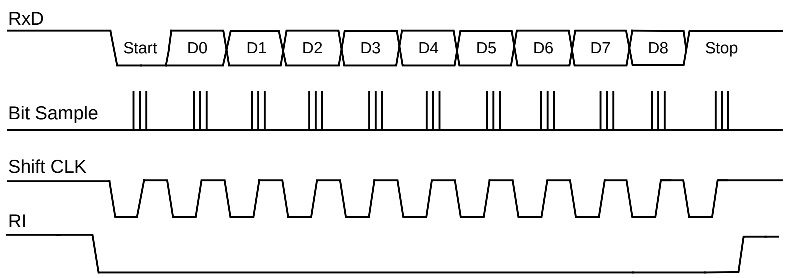
Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. 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.

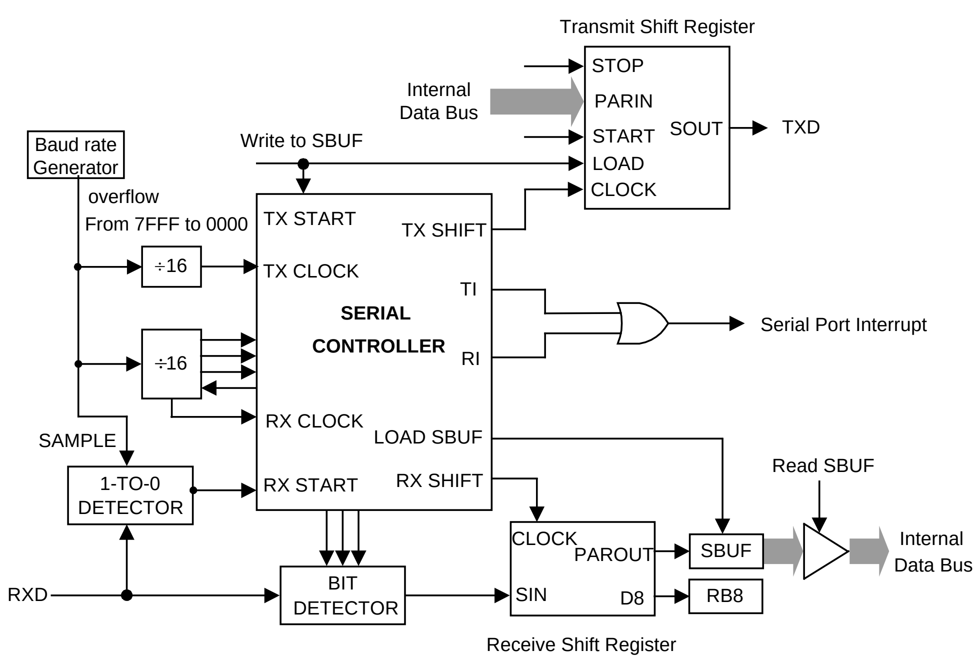
At the time, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



Receive Timing of Mode 2

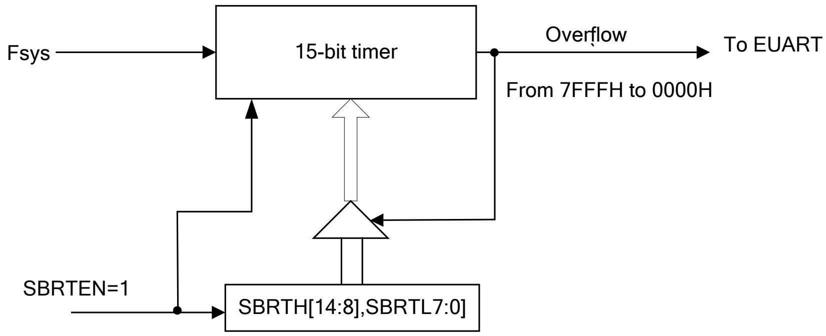
Mode3: 9-Bit EUART1, Variable Baud Rate, Asynchronous Full-Duplex

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



Adjustable Baud Rate

EUART1 has its own baud rate generator, which is actually a 15-bit up counter.



Baudrate Generator for EUART

From the figure, the baud rate generator overflow rate: $SBRToverflowrate = \frac{Fsys}{32768 - SBRT1}$, $SBRT1 = [SBRTH1, SBRTL1]$.

Therefore, EUART1 baud rate calculation formula in each mode is as follows:

In Mode0, the baud rate is programmable to be 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.

The baud rate can be adjust at a accuracy of one system clock period in Mode1 & Mode3, the formula is as follows

$$BaudRate = \frac{Fsys}{16 \times (32768 - SBRT1) + SFINE1}$$

For example: If you want to get the baud rate of 115200Hz in condition of Fsys = 8MHz, SBRT and SFINE value is calculated as follows:

8000000/16/115200 = 4.34

SBRT1 = 32768 - 4 = 32764

115200 = 8000000/(16 X 4 + SFINE1)

SFINE1 = 5.4 $\approx$ 5

The actual baud calculated according to this fine-tuning mode is 19230 with 0.16% error which will be 8.5% in the past.
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.

$$BaudRate = 2^{SMOD1} \times (\frac{f_{SYS}}{64})$$

Multi-Processor Communication

Software Address Recognition

Modes 2 and 3 of the EUART1 have a special provision for multi-processor communication. In these modes, 9 data bits are received. The 9th bit goes into RB8. Then a stop bit follows. The EUART1 can be programmed such that when the stop bit is received, the EUART1 interrupt will be activated (i.e. the request flag RI is set) only if RB8 = 1. This feature is enabled by setting the bit SM2 in SCON.

A way to use this feature in multiprocessor communications is as follows. If the master processor wants to transmit a block of data to one of the several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte.

With SM2 = 1, no other slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. After having received a complete message, the slave sets SM2 again. The slaves that were not addressed leave their SM2 set, ignoring the incoming data bytes.

Note: *In Mode0, SM2 is used to select baud rate doubling. In Mode1, SM2 can be used to check the validity of the stop bit. If SM2 = 1 in Mode1, the receive interrupt will not be activated unless a valid stop bit is received.*

Automatic (Hardware) Address Recognition

In Mode2 & 3, setting the SM2 bit will configure EUART1 act as following: when a stop bit is received, EUART1 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 EUART1 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.

	Slave 1	Slave 2
SADDR1	10100100	10100111
SADEN1 (0 mask)	11111010	11111001
Given Address	10100x0x	10100xx1
Broadcast Address	1111111x	11111111

The Given address for slave 1 and 2 differ in the LSB. For slave 1, it doesn'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 EUART1 0 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.

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.

Register

Table 8.5 Power Control

DFH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON1	SMOD1	SSTAT1	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
Reset Value (POR/WDT/LVR/PIN)	0	0	-	-	-	-	-	-

Bit Number	Bit Mnemonic	Description
7	**SMOD1**	**Baud rate doubler** 0: In mode 2, the baud rate is system clock1/64 1: In mode 2, the baud rate is system clock1/32
6	**SSTAT1**	**SCON[7:5] function select bit** 0: SCON1[7:5] operates as SM10, SM11, SM12 1: SCON1[7:5] operates as FE1, RXOV1, TXCOL1

Table 8.6 EUART1 Control & Status Register

D8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON1	SM10 /FE1	SM11 /RXOV1	SM12 /TXCOL1	REN1	TB81	RB81	TI1	RI1
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	**SM1[0:1]**	**EUART1 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	**FE1**	**EUART1 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	**RXOV1**	**EUART1 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	**SM12**	**EUART1 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	**TXCOL1**	**EUART1 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

(to be continued)

(continue)

Bit Number	Bit Mnemonic	Description
4	REN1	**EUART1 Receiver enable bit** 0: Receive Disable 1: Receive Enable
3	TB81	**The 9th bit to be transmitted in Mode2 & 3 of EUART1, set or clear by software**
2	RB81	**The 9th bit to be received in Mode1, 2 & 3 of EUART1** 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 9^{th} bit that was received
1	TI1	**Transmit interrupt flag of EUART1** 0: cleared by software 1: Set by hardware
0	RI1	**Receive interrupt flag of EUART1** 0: cleared by software 1: Set by hardware

Table 8.7 EUART1 Data Buffer Register

D9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUF1	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	SBUF1[7:0]	This SFR accesses two registers; a transmit shift register and a receive latch register A write of SBUF1 will send the byte to the transmit shift register and then initiate a transmission A read of SBUF1 returns the contents of the receive latch

Table 8.8 EUART1 Slave Address & Address Mask Register

DAH-DBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SADDR1(DAH)	SADDR1.7	SADDR1.6	SADDR1.5	SADDR1.4	SADDR1.3	SADDR1.2	SADDR1.1	SADDR1.0
SADEN1(DBH)	SADEN1.7	SADEN1.6	SADEN1.5	SADEN1.4	SADEN1.3	SADEN1.2	SADEN1.1	SADEN1.0
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	SADDR1[7:0]	**SFR SADDR1 defines the EUART1's slave address**
7-0	SADEN1[7:0]	**SFR SADEN1 is a bit mask to determine which bits of SADDR1 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 8.9 EUART1 Baud Rate Generator Register

DCH-DDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBRTH1 (DCH)	SBRTEN1	SBRT1.14	SBRT1.13	SBRT1.12	SBRT1.11	SBRT1.10	SBRT1.9	SBRT1.8
SBRTL1 (DDH)	SBRT1.7	SBRT1.6	SBRT1.5	SBRT1.4	SBRT1.3	SBRT1.2	SBRT1.1	SBRT1.0
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	**SBRTEN1**	**EUART1 Baud Rate Generator Enable bit** 0: Off(default) 1: On
6-0 **7-0**	**SBRT1[14:0]**	**EUART1 Baud Rate Generator Counter High 7-bit & Low 8-bit Register**

Table 8.10 EUART1 Baud Rate Generator Fine-Tune Register

DEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SFINE1	-	-	-	-	SFINE1.3	SFINE1.2	SFINE1.1	SFINE1.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
ReWet Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	**SFINE1[3:0]**	**EUART1 Baud Rate Generator Fine Tune Data Register**

8.2 Analog Digital Converter (ADC)

8.2.1 Feature

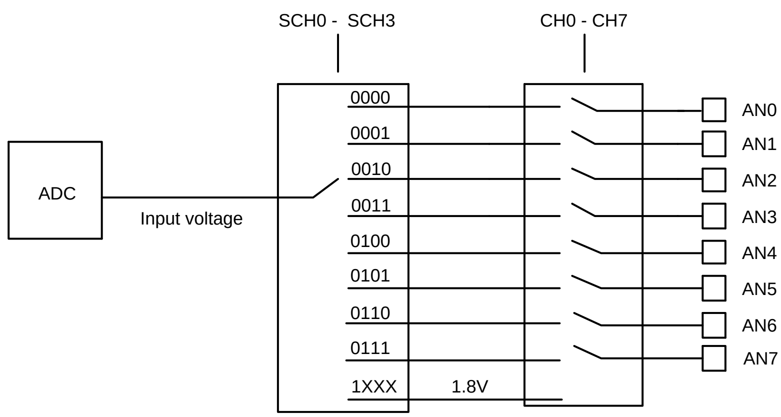
- 10-bit Resolution
- Build in V_{REF}
- 6/8 Multiplexed Input Channels

The SH79F1612B include a single ended, 10-bit SAR Analog to Digital Converter (ADC) with build in reference voltage connected to the V_{DD},The 8 ADC channels are shared with 1 ADC module; each channel can be programmed to connect with the analog input individually. Only one channel can be available at one time. GO/$\overline{\text{DONE}}$ signal is available to start convert, and indicate end of convert. When conversion is completed, the data in AD convert data register will be updated and ADCIF bit in ADCON register will be set. If ADC Interrupt is enabled, the ADC interrupt will generate.

The ADC integrates a digital compare function to compare the value of analog input with the digital value in the AD converter. If this function is enabled (set EC bit in ADCON register) and ADC module is enabled (set ADON bit in ADCON register). When the corresponding digital value of analog input is larger than the value in compare value register (ADDH/L), the ADC interrupt will occur, otherwise no interrupt will be generated. The digital comparator can work continuously when GO/$\overline{\text{DONE}}$ bit is set until software clear, which behaviors different with the AD converter operation mode.

The ADC module including digital compare module can wok in Idle mode and the ADC interrupt will wake up the Idle mode, but is disabled in Power-Down mode.

8.2.2 ADC Diagram



ADC Diagram

Note: SOP16 package does not contain AN0 and AN1 channels.

8.2.3 ADC Register

Table 8.10 ADC Control Register

93H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON	ADON	ADCIF	EC	SCH3	SCH2	SCH1	SCH0	GO/$\overline{\text{DONE}}$
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	**ADON**	**ADC Enable bit** 0: Disable the ADC module 1: Enable the ADC module
6	**ADCIF**	**ADC Interrupt Flag bit** 0: No ADC interrupt, cleared by software. 1: Set by hardware to indicate that the AD Convert has been completed, or analog input is larger than ADDH/ADDL if compare is enabled
5	**EC**	**Compare Function Enable bit** 0: Compare function disabled 1: Compare function enabled
4-1	**SCH[3:0]**	**ADC channel Select bits** (SOP16 does not contain AN0 and AN1) 0000: ADC channel AN0 0001: ADC channel AN1 0010: ADC channel AN2 0011: ADC channel AN3 0100: ADC channel AN4 0101: ADC channel AN5 0110: ADC channel AN6 0111: ADC channel AN7 1XXX:internal VCC(1.8V)
0	**GO/$\overline{\text{DONE}}$**	**ADC status flag bit** 0: Automatically cleared by hardware when AD convert is completed. Clearing this bit during converting time will stop current conversion. If Compare function is enabled, this bit will not be cleared by hardware until software clear 1: Set to start AD convert or digital compare

Table 8.11 ADC Time Control Register

94H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADT	TADC2	TADC1	TADC0	-	TS3	TS2	TS1	TS0
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
7-5	**TADC[2:0]**	**ADC Clock Period Select bits** 000: ADC Clock Period t_{AD} = 2 t_{SYS} 001: ADC Clock Period t_{AD} = 4 t_{SYS} 010: ADC Clock Period t_{AD} = 6 t_{SYS} 011: ADC Clock Period t_{AD} = 8 t_{SYS} 100: ADC Clock Period t_{AD} = 12 t_{SYS} 101: ADC Clock Period t_{AD} = 16 t_{SYS} 110: ADC Clock Period t_{AD} = 24 t_{SYS} 111: ADC Clock Period t_{AD} = 32 t_{SYS}
3-0	**TS[3:0]**	**Sample time select bits** 2 t_{AD} ≤ Sample time = (TS [3:0]+1) * t_{AD} ≤ 15 t_{AD}

Note:

(1) Make sure that t_{AD} ≧ 1μs;

(2) The minimum sample time is 2 t_{AD}, even TS[3:0] = 0000;
The maximum sample time is 15 t_{AD}, even TS[3:0] = 1111;

(3) Evaluate the series resistance connected with ADC input pin before set TS[3:0];

(4) Be sure that the series resistance connected with ADC input pin is no more than 10kΩ when 2 t_{AD} sample time is selected;

(5) Total conversion time is: 12 t_{AD} + sample time.

For Example

System Clock (SYSCLK)	TADC[2:0]	t_{AD}	TS[3:0]	Sample Time	Conversion Time
32.768kHz	000	30.5*2=61μs	0000	2*61=122μs	12*61+122=854μs
	000	30.5*2=61μs	0111	8*61=488μs	12*61+488=1220μs
	000	30.5*2=61μs	1111	15*61=915μs	12*61+915=1647μs
	111	30.5*32=976μs	0000	2*976=1952μs	12*976+1952=13664μs
	111	30.5*32=976μs	0111	8*976=7808μs	12*976+7808=19520μs
	111	30.5*32=976μs	1111	15*976=14640μs	12*976+14640=26352μs
4MHz	000	0.25*2=0.5μs	-	-	(t_{AD} <1μs, not recommended)
	001	0.25*4=1μs	0000	2*1=2μs	12*1+2=14μs
	001	0.25*4=1μs	0111	8*1=8μs	12*1+8=20μs
	001	0.25*4=1μs	1111	15*1=15μs	12*1+15=27μs
	111	0.25*32=8μs	0000	2*8=16μs	12*8+16=112μs
	111	0.25*32=8μs	0111	8*8=64μs	12*8+64=160μs
	111	0.25*32=8μs	1111	15*8=120μs	12*8+120=216μs
12MHz	000	0.083*2=0.166μs	-	-	(t_{AD} <1μs, not recommended)
	100	0.083*12=1μs	0000	2*1=2μs	12*1+2=14μs
	100	0.083*12=1μs	0111	8*1=8μs	12*1+8=20μs
	100	0.083*12=1μs	1111	15*1=15μs	12*1+15=27μs
	111	0.083*32=2.7μs	0000	2*2.7=5.4μs	12*2.7+5.4=37.8μs
	111	0.083*32=2.7μs	0111	8*2.7=21.6μs	12*2.7+21.6=54μs
	111	0.083*32=2.7μs	1111	15*2.7=40.5μs	12*2.7+40.5=72.9μs

Table 8.12 ADC Channel Configure Register

95H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCH	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
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	**CH[7:0]**	**Channel Configuration bits** (SOP16 does not contain CH0,CH1) 0: P1.0-P1.7 are I/O port 1: P1.0-P1.7 are ADC input port

Table 8.13 AD Converter Data Register (Compare Value Register)

96H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDL	-	-	-	-	-	-	A1	A0
R/W	-	-	-	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-	-	0	0
97H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
ADDH	A9	A8	A7	A6	A5	A4	A3	A2
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
1-0 **7-0**	**A9-A0**	**ADC Data register** Digital Value of sampled analog voltage, updated when conversion is completed If ADC Compare function is enabled (EC = 1), this is the value to be compared with he analog input

The Approach for AD Conversion:

(1) Select the analog input channels and reference voltage;
(2) Enable the ADC module with the selected analog channel;
(3) Set GO/$\overline{\text{DONE}}$ = 1 to start the AD conversion;
(4) Wait until GO/$\overline{\text{DONE}}$ = 0 or ADCIF = 1, if the ADC interrupt is enabled, the ADC interrupt will occur;
(5) Acquire the converted data from ADDH/ADDL;
(6) Repeat step 3-5 if another conversion is required.

The Approach for Digital Compare Function:

(1) Select the analog input channels and reference voltage;
(2) Set ADDH/ADDL to the compare value;
(3) Set EC = 1 to enable compare function;
(4) Enable the ADC module with the selected analog channel;
(5) Set GO/$\overline{\text{DONE}}$ = 1 to start the compare function;
(6) If the analog input is lager than compare value set in ADDH/ADDL, the ADCIF will be set to 1. if the ADC interrupt is enabled, the ADC interrupt will occur;
(7) The compare function will continue work until the GO/$\overline{\text{DONE}}$ bit is cleared to 0.

8.3 PWM Module

8.3.1 Feature

- 8-bit PWM modules
- Provided interrupt function on period and duty overflow
- Selectable output polarity

The SH79F1612B has a 8-bit PWM modules. The PWM modules can provide the pulse width modulation waveform with the period and the duty being controlled, individually. The PWMCON is used to control the PWM module operation with proper clocks. The PWMP is used to control the period cycle of the PWM module output. And the PWMD is used to control the duty in the waveform of the PWM module output.

8.3.2 Register

Table 8.14 PWM Control Register

D1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON	PWMEN	PWMS	PWMCK1	PWMCK0	-	-	PWMIF	PWMSS
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	**PWMEN**	**PWM Enable** 0: Disable PWM module 1: Enable PWM module
6	**PWMS**	**PWM output Polarity Selection** 0: High Active, PWMN output high during duty time, output low during remain period time 1: Low Active, PWMN output low during duty time, output high during remain period time
5-4	**PWMCK[1:0]**	**PWM clock select bit** 00: System clock/2 01: System clock/4 10: System clock/8 11: System clock/16
1	**PWMIF**	**PWM interrupt flag** 0: PWM period counter not overflow 1: Set by hardware to indicate that the PWM period counter overflow
0	**PWMSS**	**PWM output share selection** 0: PWM output disable, used as I/O port 1: PWM output enable

Table 8.15 PWM Period Control Register

D2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMP	PWMP.7	PWMP.6	PWMP.5	PWMP.4	PWMP.3	PWMP.2	PWMP.1	PWMP.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	**PWMP[7:0]**	**PWM output period cycle = PWMP * PWM clock** When PWMP = 00H, PWM pin outputs GND if the PWMS = 0 When PWMP = 00H, PWM pin outputs HIGH if the PWMS = 1

Table 8.16 PWM Duty Control Register

D3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMD	PWMD.7	PWMD.6	PWMD.5	PWMD.4	PWMD.3	PWMD.2	PWMD.1	PWMD.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	**PWMD[7:0]**	**PWM Duty cycle control, which the controls the first half time of PWM waveform** 1. If PWMP ≤ PWMD, PWM pin outputs high level when the PWMS = 0 If PWMP ≤ PWMD, PWM pin outputs low level when the PWMS = 1 2. When PWMD = 00H, PWM pin outputs GND if the PWMS = 0 When PWMD = 00H, PWM pin outputs HIGH if the PWMS = 1

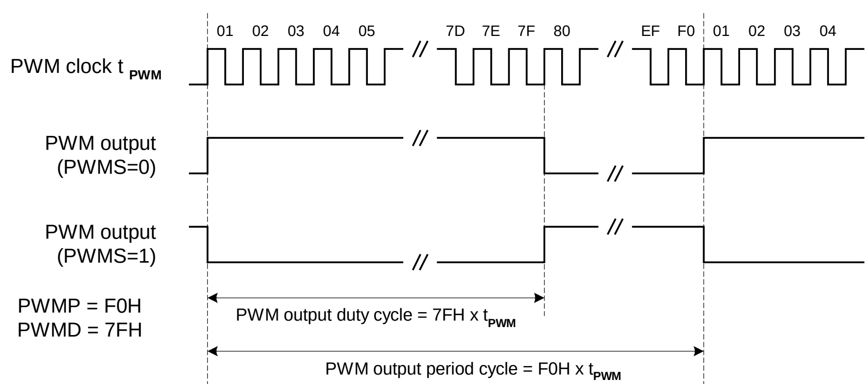
Notes:

(1) PWMEN bit can enable the PWM module.

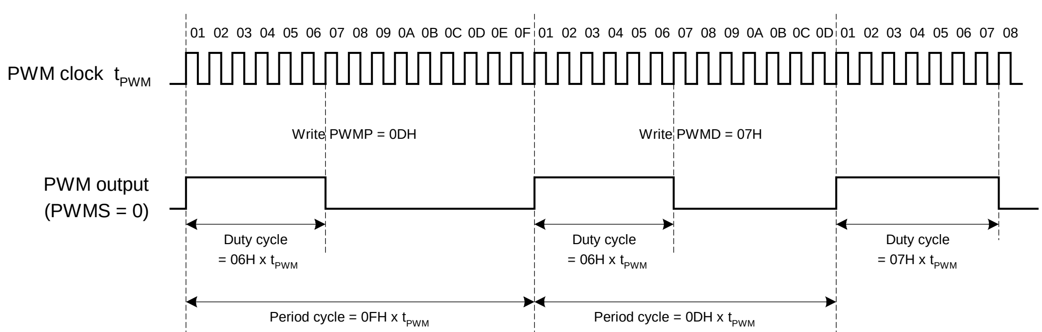
(2) PWMSS bit is used to select the P3.5 used as I/O port or PWM output.

(3) EPWM in IEN1 register can enable the PWM Timer interrupt.

(4) The PWM timer is control by PWMEN bit. If this bit is set to 1, but the PWMSS bit is cleared to '0', the PWM module used as an 8-bit Timer, if the EPWM bit in IEN1 register is set to '1', the interrupt also can be generated.



PWM Output Example



PWM Output Period or Duty Cycle Changing Example

8.4 Low Voltage Reset (LVR)

8.4.1 Feature

- Enabled by the code option and V_{LVR} is 4.1V, 3.7V or 2.8V
- LVR de-bounce timer T_{LVR} is about 30-100μs
- An internal reset flag indicates low voltage reset generates

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 30-100μs.

The LVR circuit has the following functions when the LVR function is enabled: (t means the time of the supply voltage below V_{LVR})

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

Cancels the system reset when $V_{DD} > V_{LVR}$ or $V_{DD} < V_{LVR}$, but $t < T_{LVR}$.

The LVR function is enabled by the code option.

It is typically used in AC line or large battery supplier applications, where heavy loads may be switched on and cause the MCU supply-voltage temporarily falls below the minimum specified operating voltage. This feature can protect system from working under bad power supply environment.

8.5 Watchdog Timer (WDT) and Reset State

8.5.1 Feature

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

OVL Reset

To enhance the anti-noise ability, SH79F1612B 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:

8.5.2 Register

Table 8.17 Reset Control Register

B1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RSTSTAT	WDOF	-	PORF	LVRF	CLRF	WDT.2	WDT.1	WDT.0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR)	0	-	1	0	0	0	0	0
Reset Value (WDT)	1	-	u	u	u	0	0	0
Reset Value (LVR)	u	-	u	1	u	0	0	0
Reset Value (PIN)	u	-	u	u	1	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 and 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
3	**CLRF**	**Pin Reset Flag** Set only by pin reset, cleared by software or Power On Reset 0: No Pin Reset occurs 1: Pin Reset occurred
2-0	**WDT[2:0]**	**WDT Overflow period control bit** 000: Overflow period minimal value = 4096ms 001: Overflow period minimal value = 1024ms 010: Overflow period minimal value = 256ms 011: Overflow period minimal value = 128ms 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 value list above.*

8.6 Power Management

8.6.1 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, SH79F1612B supplies two power saving modes: Idle mode and Power-Down mode. These two modes are controlled by PCON & SUSLO register.

8.6.2 Idle Mode

In this mode, the clock to 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 SH79F1612B 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 to 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 high on the RESET pin, WDT RESET if enabled, LVR REST if enabled), this will restore the clock to the CPU, the SUSLO register and the IDL bit in PCON register will be cleared by hardware, finally the SH79F1612B 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.

8.6.3 Power-Down Mode

The Power-Down mode places the SH79F1612B in a very low power state. Power-Down mode will stop all the clocks including CPU and peripherals. 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 SH79F1612B 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 SH79F1612B 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 two ways to exit the Power-Down mode:

(1) An active external Interrupt such as INT0, INT1 & INT2 will make SH79F1612B exit Power-Down mode. The oscillator will start after interrupt happens, after warm-up time, the clocks to 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, program execution resumes with the instruction immediately following the instruction that activated Power-Down mode.

(2) Reset signal (logic high on the RESET pin, WDT RESET if enabled, LVR REST if enabled). This will restore the clock to the CPU after warm-up time, the SUSLO register and the PD bit in PCON register will be cleared by hardware, finally the SH79F1612B 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.*

8.6.4 Register

Table 8.18 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 double bit**
6	**SSTAT**	**SCON[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 8.19 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
```

8.7 Warm-up Timer

8.7.1 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 startup

SH79F1612B 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 customer option etc.

SH79F1612B 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 Power-down mode.

After power-on, SH79F1612B will start power warm-up procedure first, and then oscillator warm-up procedure.

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	1000CKs	NO	1000 CKs	YES	64CKs	YES

OSC Warm-up Time

Option: OP_WMT / Oscillator Type	**00**	**01**	**10**	**11**
Ceramic	2^{13} X T_{OSC}	2^{11} X T_{OSC}	2^{9} X T_{OSC}	2^{7} X T_{OSC}
Crystal	2^{17} X T_{OSC}	2^{15} X T_{OSC}	2^{13} X T_{OSC}	2^{11} X T_{OSC}
32kHz Crystal	2^{13} X T_{OSC}			
Internal RC	2^{7} X T_{OSC}			

8.8 Low Power Detect (LPD)

8.8.1 Feature

- An internal flag indicates low power is detected
- 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 8.20 Low Power Detection Control Register

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

**: LPDF can be cleared by software only.*

: **Program Note:

If LPD detect voltage is select as Vdd, and P1.3 used as analog input port (ADC Channel), it can be set by OP_LPDFLAG in Option to choose whether to set LPD flag when LPD happend.

If OP_LPDFLAG = 0, LPD flag bit can not be set when LPD happened;

If OP_LPDFLAG = 1, LPD flag bit can be set when LPD happened.

Bit Number	Bit Mnemonic	Description
7	**LPDEN**	**LPD Enable bit** 0: Disable lower power detection 1: Enable lower power detection
6	**LPDF**	**LPD Flag bit** 0: No LPD happened, clear by hardware, means current voltage is above LPD value in LPDS [1:0] 1: LPD happened, set by hardware, means current voltage is below LPD value in LPDS [1:0]
5	**LPDV**	**LPD Detect Select bit** 0: detect Vdd 1: detect VLPD (P1.3) pin voltage
2-0	**LPDS[2:0]**	**LPD Voltage Select bit** 000:3.7V 001:3.9V 010:4.2V 011:4.4V 100:2.9V 101:3.1V 110:3.3V 111:3.5V

8.9 Code Option

OP_WDT:

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

OP_WDTPD:

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

OP_WMT: (unavailable for 32kHz crystal and Internal RC)

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

OP_OSC:

000: Internal RC (16.6MHz) (Default)
010: External clock (30kHz - 16.6MHz)
011: 32.768kHz crystal oscillator and Internal RC16.6M (open by the instructions)
101: Crystal oscillator (4MHz - 16.6MHz)
110: Ceramic resonator (2MHz - 16.6MHz)
Others: Internal RC (16.6MHz)

OP_RST:

0: P4.0 used as RST pin (Default)
1: P4.0 used as I/O pin

OP_LVREN:

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

OP_LVRLE:

00: 4.1V LVR level 1 (Default)
10: 3.7V LVR level 2
01: 2.8V LVR level 3

OP_SCM:

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

OP_IO:

0: IO structure is only input mode after power-on reset
1: IO structure is Quasi-Bi mode after power-on reset (Default)

OP_OSCDRV:

10: External oscillator drive capability: Middle (Default)
01: External oscillator drive capability: Maximum

OP_LPDFLAG[4]:

0:LPD FLAG cannot be set when P1.3 used as analog input pin and LPD detect voltage is select as VDD (Default)
1:LPD FLAG can be set when P1.3 used as analog input pin and LPD detect voltage is select as VDD.

OP_P3:

0: P3 sink ability normal mode. (Default)
1: P3 sink ability large mode

OP_P1P4:

0: P1P4 drive ability normal mode. (Default)
1: P1P4 drive ability large mode

OP_OSC and OP_OSCDRV are used as following combination:

NO.	OP_OSC	OP_OSCDRV	Oscillate Type
1	101 (Crystal oscillator)	01 (Middle)	Crystal oscillator 4MHz - 16MHz
2	110 (Ceramic resonator)	01 (Middle)	Ceramic resonator 2MHz - 8MHz
3	110 (Ceramic resonator)	10 (Maximum)	Ceramic resonator 8MHz - 16MHz

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°C to +85°C

Storage Temperature. . . . .-55℃ to +125℃

***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} = 2.8 - 5.5V, GND = 0V, T_A = 25°C, unless otherwise specified)

Parameter	Symbol	Min.	Typ.*	Max.	Unit	Condition
Operating Voltage	V_{DD}	2.8	5.0	5.5	V	32.768kHz ≤ f_{OSC} ≤ 16.6MHz
Operating Current	I_{OP}	-	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), all other function block off
Stand by Current (IDLE)	I_{SB1}	-	25	35	μA	f_{OSC} = 32.768kHz, V_{DD} = 5.0V All output pins unload (including all digital input pins unfloating), all other function block off
	I_{SB2}	-	3	5	mA	f_{OSC} = 16.6MHz, V_{DD} = 5.0V All output pins unload (including all digital input pins unfloating) ,all other function block off
Stand by Current (Power-Down)	I_{SB3}	-	-	15	μA	f_{OSC} = 16.6MHz, V_{DD} = 5.0V All output pins unload(including all digital input pins unfloating), CPU off (Power-Down), LVR off, LCD off, WDT off, all other function block off
WDT Current	I_{WDT}	-	1	3	μA	All output pins unload, WDT on, V_{DD} = 5.0V
LPD Current	I_{LPD}	-	3	5	μA	V_{DD} = 5.0V
Input Low Voltage	V_{IL}	GND	-	0.2 X V_{DD}	V	I/O Ports(all pin have schmitt trigger)
Input High Voltage	V_{IH}	0.8 X V_{DD}	-	V_{DD}	V	I/O Ports(all pin have schmitt trigger)
Input Leakage Current	I_{IL}	-1	-	1	μA	Input pad, V_{IN} = V_{DD} or GND
Output Leakage Current	I_{OL}	-1	-	1	μA	Open-drain, V_{out} = V_{DD} or GND
Very weak Pull-high Resistor	R_{PH1}	-	300	-	kΩ	V_{DD} = 5.0V, V_{IN} = GND
Weak Pull-high Resistor	R_{PH2}	-	10	-	kΩ	V_{DD} = 5.0V, V_{IN} = GND
Output High Voltage1	V_{OH1}	V_{DD} - 0.7	-	-	V	I/O Ports (P1, P3 ,P4), I_{OH} = -10mA, V_{DD} = 5.0V, (push-pull mode, customer option is off)
Output High Voltage2	V_{OH2}	V_{DD} - 0.7	-	-	V	I/O Ports (P1, P4), I_{OH} = -15mA, V_{DD} = 5.0V, (push-pull mode, customer option is off)
Output Low Voltage	V_{OL}	-	-	GND + 0.6	V	I/O Ports, I_{OL} = 25mA, V_{DD} = 5.0V, push-pull mode
High Drive Ports Sink Current capacity	IOL	80	100	-	mA	I/O Ports (P3), V_{DD} = 5.0V, V_{OL} = GND + 1.5V, (push-pull mode, customer option is off)

Note:

(1) "" Data in "Typ." Column is at 5.0V, 25°C, unless otherwise specified.*

(2) Maximum value of the supply current to V_{DD} is 100mA.

(3) Maximum value of the output current from GND is 150mA.

A/D Converter Electrical Characteristics (T_A = -25°C, Unless otherwise specified)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Supply Voltage	V_{AD}	4.5	5.0	5.5	V	
Resolution	N_R	-	10	-	bit	GND ≤ V_{AIN} ≤ V_{REF}
A/D Input Voltage	V_{AIN}	GND	-	-	V	
A/D Input Resistor*	R_{AIN}	2	-	-	MΩ	V_{IN} = 5.0V
Recommended impedance of analog voltage source**	Z_{AIN}	-	-	10	kΩ	
A/D conversion current	I_{AD}	-	1	3	mA	ADC module operating, V_{DD} = 5.0V
A/D Input current	I_{ADIN}	-	-	10	μA	V_{DD} = 5.0V
Differential linearity error	D_{LE}	-	-	±1	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Integral linearity error	I_{LE}	-	-	±2	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Full scale error	E_F	-	±1	±3	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Offset error	E_Z	-	±0.5	±2	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Total Absolute error	E_{AD}	-	-	±3	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Total Conversion time**	TCON	14	-	-	μs	10 bit Resolution, V_{DD} = 5.0V

Note:
(1) "" Here the A/D input Resistor is the DC input-resistance of A/D itself.*
*(2) "**" Be sure that the series resistance connected with ADC input pin is no more than 10kΩ.*

AC Electrical Characteristics (V_{DD} = 2.8V - 5.5V, GND = 0V, T_A = 25°C, f_{OSC} = 30KHz - 16.6MHz, unless otherwise specified)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Oscillator start time1	T_{OSC1}	-	1	2	s	32.768kHz
Oscillator start time2	T_{OSC2}	-	-	2	ms	16.6MHz
RESET pulse width	t_{RESET}	10	-	-	μs	High active
RESET Pull-high Resistor	R_{RPH}	-	30	-	kΩ	V_{DD} = 5.0V, V_{IN} = GND
Frequency Stability (RC) *	F_{RC}	-	-	±2	%	RC Oscillator |F - 16.6MHz|/16.6MHz (V_{DD} = 2.8 - 5.5V, T_A = -40°C ~ +85°C)

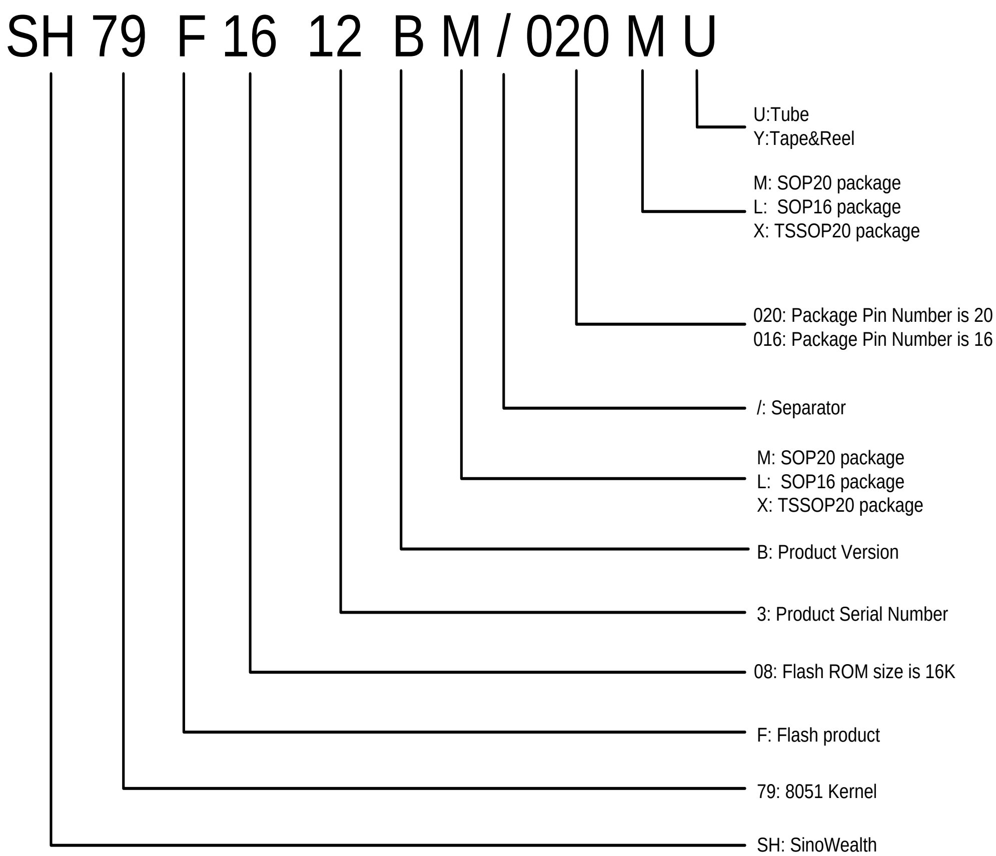
Low Voltage Reset Electrical Characteristics (V_{DD} = 2.8V - 5.5V, GND = 0V, T_A = +25°C, unless otherwise specified)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
LVR Voltage1	V_{LVR1}	3.95	4.1	4.25	V	LVR1 enabled, V_{DD} = 2.8V - 5.5V
LVR Voltage2	V_{LVR2}	3.55	3.7	3.85	V	LVR2 enabled, V_{DD} = 2.8V - 5.5V
LVR Voltage3	V_{LVR2}	2.7	2.8	2.9	V	LVR2 enabled, V_{DD} = 2.8V - 5.5V

11. Ordering Information

Part No.	Package
SH79F1612BL/016LU	SOP16
SH79F1612BM/020MU	SOP20
SH79F1612BM/020MY	SOP20
SH79F1612BX/020XU	TSSOP20
SH79F1612BX/020XY	TSSOP20

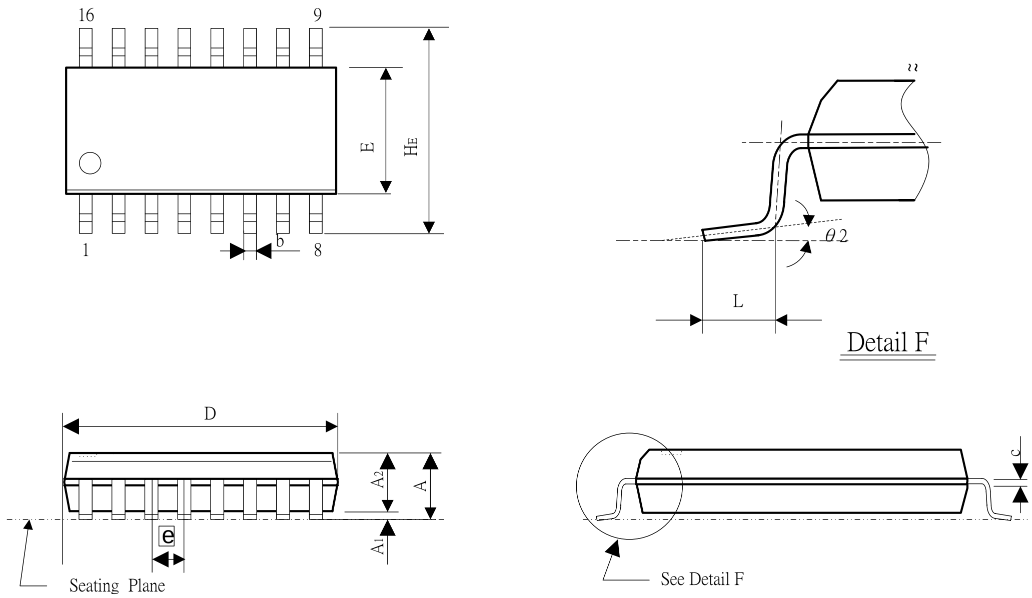
12. Product Identification System



13. Package Information

SOP 16L(150mil) Outline Dimensions

unit: inches/mm

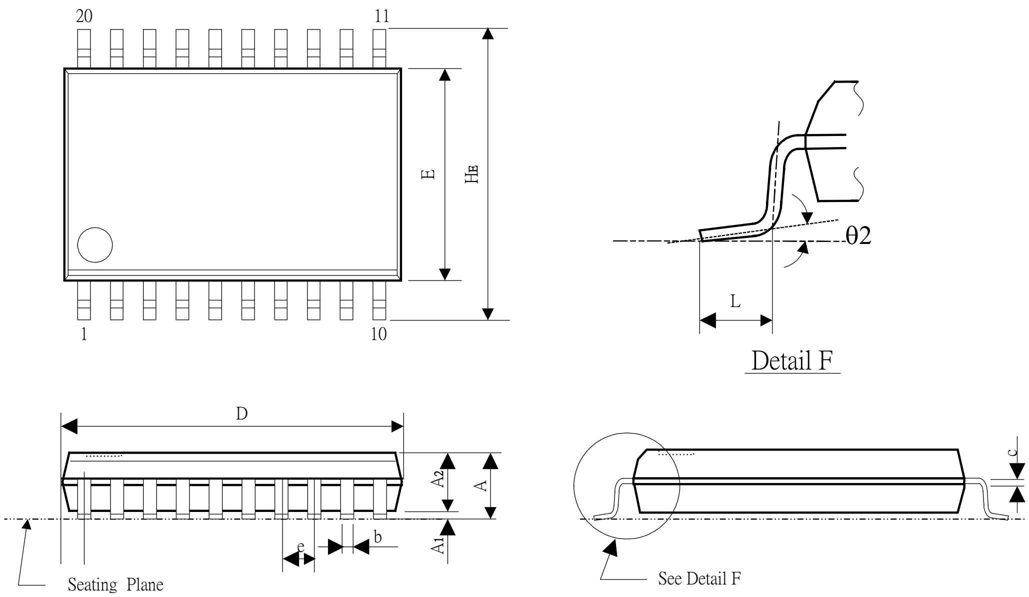


Symbol	Dimensions in inches		Dimensions in mm	
	Min	Max	Min	Max
A	0.053	0.071	1.35	1.8
A1	0.004	0.010	0.1	0.25
A2	0.049	0.061	1.25	1.55
b	0.013	0.020	0.33	0.51
c	0.008	0.014	0.2	0.35
D	0.386	0.402	9.8	10.2
E	0.150	0.157	3.8	4
e	0.050 (BSC)		1.27 (BSC)	
H_E	0.228	0.248	5.8	6.3
L	0.016	0.050	0.4	1.27
θ2	0°	8°	0°	8°

Notice:

(1) Both package length and width do not include mold flash.
(2) Tolerance is ±0.1mm if not specified.
(3) Coplanarity: 0.1mm max.
(4) Controlling dimension: mm.

SOP20L Outline Dimensions unit: inches/mm



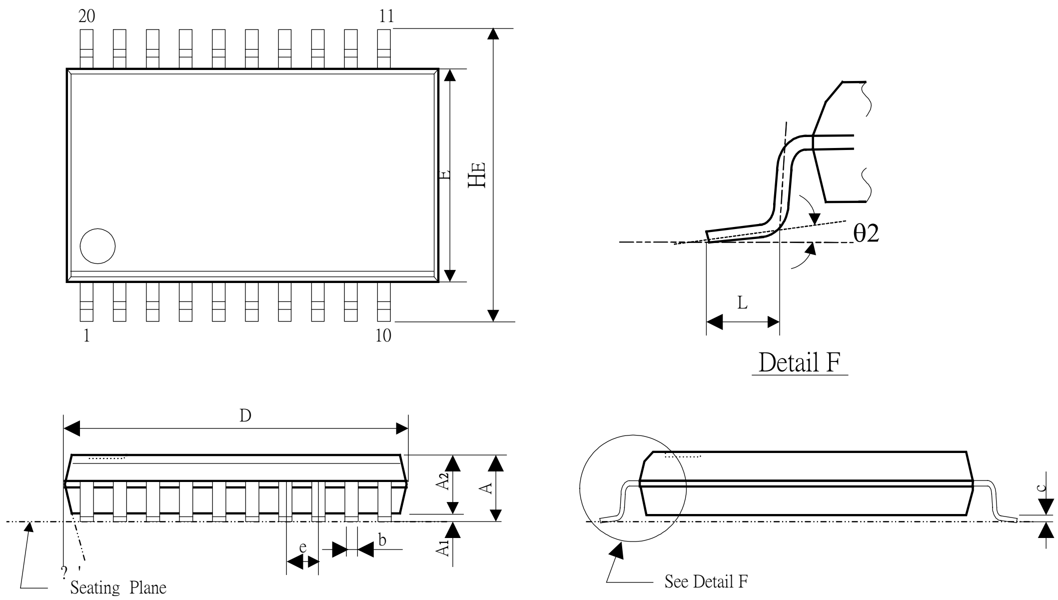
Symbol	Dimensions in inches		Dimensions in mm	
	Min	Max	Min	Max
A	0.093	0.104	2.35	2.65
A1	0.004	0.012	0.10	0.30
A2	0.083	0.098	2.10	2.50
b	0.013	0.020	0.33	0.51
c	0.008	0.013	0.20	0.33
D	0.493	0.516	12.52	13.10
E	0.291	0.299	7.40	7.60
e	0.050 (BSC)		1.27 (BSC)	
H_E	0.398	0.418	10.11	10.61
L	0.016	0.050	0.40	1.27
θ2	0°	8°	0°	8°

Notice:

(1) Both package length and width do not include mold flash.
(2) Tolerance is ±0.1mm if not specified.
(3) Coplanarity: 0.1mm max.
(4) Controlling dimension: mm.

TSSOP 20L Outline Dimensions

unit: inches/mm



Symbol	Dimensions in inches		Dimensions in mm	
	MIN	MAX	MIN	MAX
A	---	0.048	---	1.2
A**1**	0.002	0.006	0.05	0.15
A**2**	0.031	0.041	0.8	1.05
b	0.007	0.012	0.18	0.3
C	0.004	0.008	0.09	0.2
D	0.252	0.26	6.4	6.6
E	0.169	0.177	4.3	4.5
HE	0.246	0.258	6.25	6.55
e	0.026 (BSC)		0.65 (BSC)	
L	0.018	0.030	0.45	0.75
θ2	0°	8°	0°	8°

Notice:

(1) Both package length and width do not include mold flash.
(2) Tolerance is ±0.1mm if not specified.
(3) Coplanarity: 0.1mm max.
(4) Controlling dimension: mm.

14. Product SPEC. Change Notice

Version	Content	Date
2.2	Original	May. 2021

IMPORTANT NOTICE

Content

1. FEATURES 1
2. GENERAL DESCRIPTION 1
3. BLOCK DIAGRAM 2
4. PIN CONFIGURATION 3
4.1 SOP 16PIN 3
4.2 SOP/TSSOP 20PIN 3
5. PIN DESCRIPTION 5
6. SFR MAPPING 6
7. NORMAL FUNCTION 12
7.1 CPU 12
7.1.1 CPU Core SFR 12
7.1.2 Enhanced CPU core SFRs 13
7.1.3 Register 13
7.2 RAM 14
7.2.1 Feature 14
7.2.2 Register 14
7.3 FLASH PROGRAM MEMORY 15
7.3.1 Feature 15
7.3.2 Flash Operation in ICP Mode 17
7.4 SSP FUNCTION 18
7.4.1 SSP Register 18
7.4.2 Flash Control Flow 20
7.4.3 SSP Programming Notice 21
7.4.4 Readable Random Code 21
7.5 SYSTEM CLOCK AND OSCILLATOR 22
7.5.1 Feature 22
7.5.2 Clock Definition 22
7.5.3 Description 22
7.5.4 Register 22
7.5.5 Oscillator Type 24
7.5.6 Capacitor Selection for Oscillator 25
7.6 SYSTEM CLOCK MONITOR (SCM) 26
7.7 I/O PORT 27
7.7.1 Feature 27
7.7.2 Register 27
7.7.3 Port Structure 28
7.7.4 Port Share 30
7.8 TIMER 32
7.8.1 Feature 32
7.8.2 Timer0/1 32
7.8.3 Timer2 37
7.9 INTERRUPT 43
7.9.1 Feature 43
7.9.2 Description 43
7.9.3 Interrupt Enable Control 43
7.9.4 Interrupt Flag 45
7.9.5 Interrupt Vector 47
7.9.6 Interrupt Priority 47
7.9.7 Interrupt Handling 48
7.9.8 Interrupt Response Time 48
7.9.9 External Interrupt Inputs 49
7.9.10 Interrupt Summary 49
8. ENHANCED FUCNTION 50
8.1 EUART/EUART1 50
8.1.1 Feature 50
8.1.2 EUART Mode Description 50
8.1.3 Baud Rate Generate 55
8.1.4 Multi-Processor Communication 55
8.1.5 Error Detection 56
8.1.6 Register 57

8.1.7 EUART1 ..... *59*
8.2 ANALOG DIGITAL CONVERTER (ADC) 69
8.2.1 Feature *69*
8.2.2 ADC Diagram *69*
8.2.3 ADC Register *70*
8.3 PWM MODULE 73
8.3.1 Feature *73*
8.3.2 Register *73*
8.4 LOW VOLTAGE RESET (LVR) 75
8.4.1 Feature *75*
8.5 WATCHDOG TIMER (WDT) AND RESET STATE 76
8.5.1 Feature *76*
8.5.2 Register *76*
8.6 POWER MANAGEMENT 77
8.6.1 Feature *77*
8.6.2 Idle Mode *77*
8.6.3 Power-Down Mode *77*
8.6.4 Register *78*
8.7 WARM-UP TIMER 79
8.7.1 Feature *79*
8.8 LOW POWER DETECT (LPD) 80
8.8.1 Feature *80*
8.8.2 Register *80*
8.9 CODE OPTION 81
9. INSTRUCTION SET 82
10. ELECTRICAL CHARACTERISTICS 87
11. ORDERING INFORMATION 89
12. PRODUCT IDENTIFICATION SYSTEM 90
13. PACKAGE INFORMATION 91
14. PRODUCT SPEC. CHANGE NOTICE 94