STC11F02 Family
STC11F01E/02E/04E/06

Features

- High Performance Enhanced 80C51 Unit.
- Operating voltage range: 3.3V / 5.0V
- Operating frequency range: **35**Mhz(Max).
- On-chip 6K(max) FLASH program memory with flexible ISP/IAP capability
- On-chip 128+128 byte scratch-pad RAM.
- Code protection for flash memory access
- Two 16-bit timer/counter(T0/T1)
- 6 vector-address, 4 level priority interrupt capability
- One enhanced UART with hardware address-recognition and frame-error detection function
- One 15 bits Watch-Dog-Timer with 8-bit pre-scalar (one-time-enabled)
- Build In internal 6MHz RC oscillator
- Three power management modes: idle mode, slow down mode and power-down mode

  Power down mode can be woken-up by any external interrupt pins and any RXD interrupt pins.
- Maximum 16 programmable I/O ports are available
- Package type : PDIP-16/18/20 SOP-16/18/20

General Description

STC11Fxx is a single-chip 8-bit micro-controller with instruction sets fully compatible with industrial-standard 80C51 series micro controller.

There is very excellent MCU kernel built in this device compared to general 80C51 MCUs those take twelve oscillating cycles to finish an instruction, the device could take only one oscillating cycle to finish one instruction.

There is 6K(max) bytes flash memory embedded which could be used as program or data. Also the In-System Programming and In-Application Programming mechanisms are supported. The data endurance of the embedded flash gets over 20,000 times, and 21 years data retention is guaranteed.

The operation frequency reaches at 35MHzs. An user can apply a crystal oscillator for the oscillating source, or alternatively uses the built in 6MHz RC oscillator to save system cost.

The UART interfaces make the device convenient to communicate with the peripheral component, say talking to a personal computer via RS-232 port, or communicating with a serial memory.

Up to 16 programmable GPIOs are available from STC11Fxx.

The STC11Fxx is really the most efficient MCU adapted for simple control; say electronic scales, remote controller, security encoder/decoder, Video Player Controller, and user interface controller.

Order Information:

Part Number	Temperature Range	Package	Packing	Operation Voltage
STC11Fxx-PDIP	**Industrial**	**PDIP-16/18/20**	**Tube**	**3.3V / 5.0V**
STC11Fxx-SOP	**Industrial**	**SOP-16/18/20**	**Tube**	**3.3V / 5.0V**

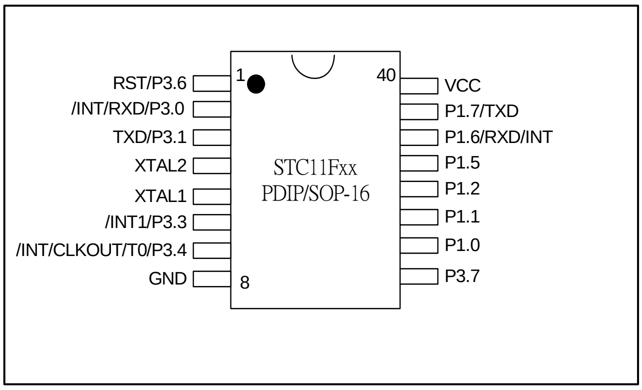
.x: voltage aa: rom size bbb:ADC,PWM .or. none cc:active frequency
.d: temperature “I” for industrial eeee: package type ff: pin count

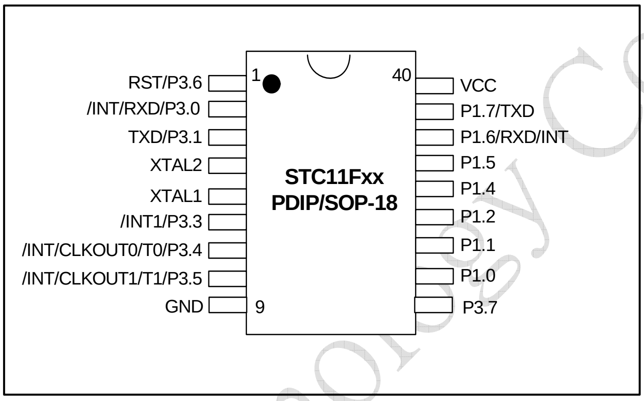
Pin Description

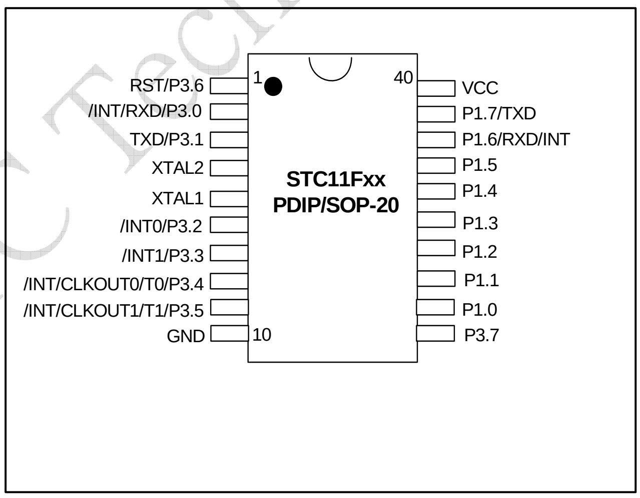
Pin Definition

MNEMONIC	Package Type			DESCRIPTION
	PDIP16	PDIP-18	PDIP-20	
P1.0 P1.1 P1.2 P1.3 P1.4 P1.5 P1.6/INT P1.7/TXD	10 11 12 - - 13 14 15	11 12 13 - 14 15 16 17	12 13 14 15 16 17 18 19	**Port1**: General-purposed I/O with weak pull-up resistance inside. When 1s are written into Port1, the strong output driving PMOS only turn-on two period and then the weak pull-up resistance keep the port high.
P3.0/RXD/INT P3.1/TXD P3.2/INT0 P3.3/INT1 P3.4/CLKOUT0 P3.5/CLKOUT1 P3.6/RST P3.7/RD	2 3 - 6 7 - - -	2 3 - 6 7 8 1 10	2 3 6 7 8 9 1 11	**Port3**: General-purposed I/O with weak pull-up resistance inside. When 1s are written into Port1, the strong output driving PMOS only turn-on two period and then the weak pull-up resistance keep the port high. Port3 also serves the special function of STC11Fxx.
RESET	1	1	1	**RESET**: A high on this pin for at least two machine cycles will reset the device.
XTAL1	5	5	5	**Crystal1**: Input to the inverting oscillator amplifier.
XTAL2	4	4	4	**Crystal2**: Output from the inverting amplifier.
VDD	16	18	20	**Power**
GND	8	9	10	**Ground**

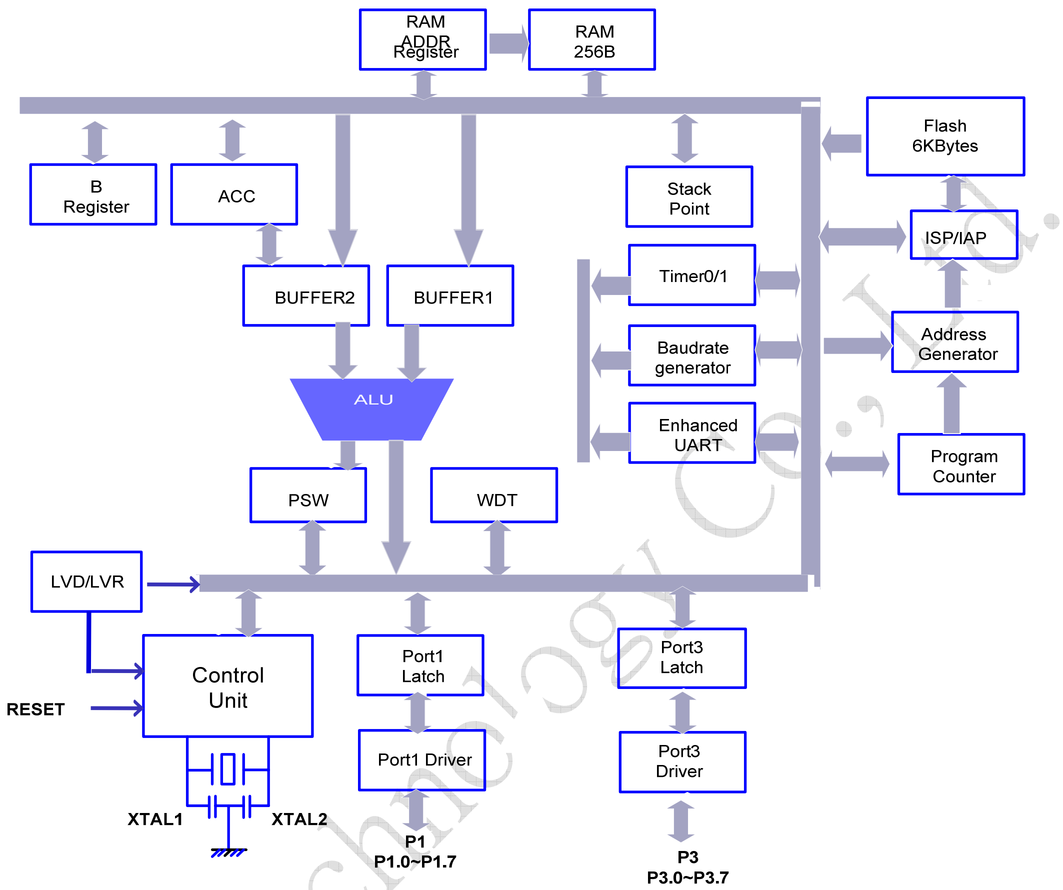
Pin Configuration







Block Diagram



STC11Fxx Block Diagram

Special Function Register

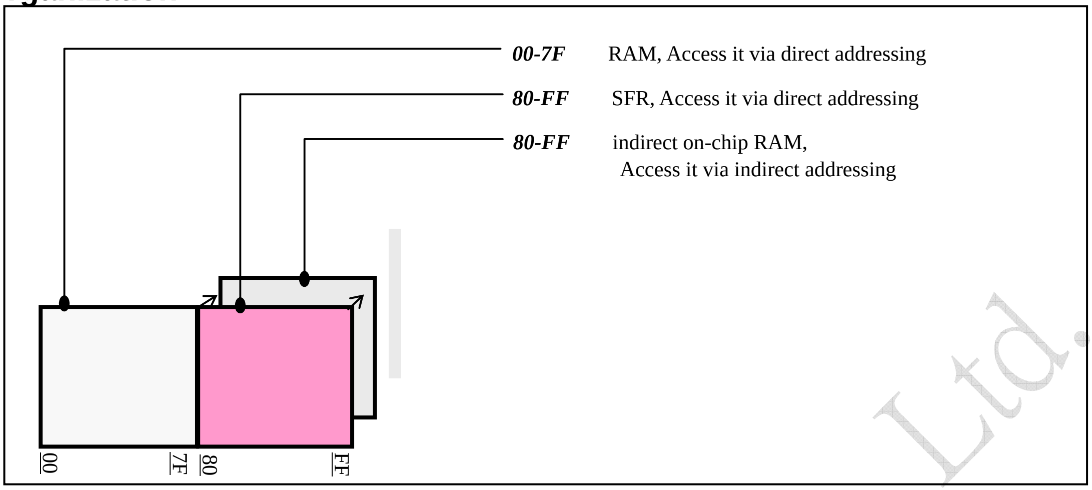
Address Map

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
0F8H									0FFH
0F0H	B 000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H		WDT_CONTR 0x000000	IAP_DATA 11111111	IAP_ADDRH 00000000	IAP_ADDRL 00000000	IAP_CMD xxxxxx00	IAP_TRIG xxxxxxxx	IAP_CONTR 0000x000	0C7H
0B8H	IP xx000000	SADEN 00000000							0BFH
0B0H	P3 11111111	P3M1 00000000	P3M0 00000000						0B7H
0A8H	IE 0x000000	SADDR 00000000	WKTCL 00000000	WKTCH 00000000					0AFH
0A0H	P2 11111111		AUXR1 Xxxx0xx0						0A7H
098H	SCON 00000000	SBUF xxxxxxxx			BRT 00000000				09FH
090H	P1 11111111	P1M1 00000000	P1M0 00000000	P0M1 00000000	P0M0 00000000	P2M1 00000000	P2M0 00000000	CLK_DIV Xxxxx000	097H
088H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR 0000x000	WAKE_CLKO x000x000	08FH
080H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000	SPISTAT 00xxxxxx	SPICTL 00000100	SPIDAT 00000000	PCON 00110000	087H
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

Bits Description

SYMBOL	DESCRIPTION	ADDR	MSB			BIT ADDRESS AND SYMBOL				LSB	INITIAL VALUE
P0	Port 0	80H	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111B
SP	Stack Pointer	81H									00000111B
DPL	Data Pointer Low	82H									00000000B
DPH	Data Pointer High	83H									00000000B
PCON	Power Control	87H	SMOD	SMOD0	-	POF	GF1	GF0	PD	IDL	00010000B
TCON	Timer Control	88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000B
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00000000B
TL0	Timer Low 0	8AH									00000000B
TL1	Timer Low 1	8BH									00000000B
TH0	Timer High 0	8CH									00000000B
TH1	Timer High 1	8DH									00000000B
AUXR	Auxiliary register	8EH	T0X12	T1X12	UART_ M0X6	BRTR	-	BRTR X12-		S1BRS	0000x000B
WAKE_ CLK0		8FH		RXD_PIN_ IE	T1_PIN_ IE	T0_PIN_ IE		BRTCLKO	T1CLKO	T0_CLKO	X000x000B
P1	Port 1	90H	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111B
P1M1	P1 configuration 0	91H									00000000B
P1M0	P1 configuration 1	92H									00000000B
P0M1	P0 configuration 0	93H	-	-	-	-					00000000B
P0M0	P0 configuration 1	94H	-	-	-	-					00000000B
P2M1	P2 configuration 0	95H									00000000B
P2M0	P2 configuration 1	96H									00000000B
CLK_DIV	Clock Dvder	97H						CLKS2	CLKS1	CLKS0	Xxxxx000B
SCON	Serial Control	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	00000000B
SBUF	Serial Buffer	99H									xxxxxxxxB
BRT	Dedicate Baud-Rate Time	9CH									00000000B
P2	Port 2	A0H									11111111B
AUXR1		A2H	UART_P1				GF2			DPS	0xxx0xx0
IE	Interrupt Enable	A8H	EA	-	ET2	ES	ET1	EX1	ET0	EX0	0x000000B
SADDR	Slave Address	A9H									00000000B
WKTCL	Wake Up Control Register Low	AAH	-	-							00000000B
WAKCH	Wake Up Control Register High	ABH	WKTEN								0xxx0000B
P3	Port 3	B0H	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	11111111B
P3M1	P3 configuration 0	B1H									00000000B
P3M0	P3 configuration 1	B2H									00000000B
IP	Interrupt Priority Low	B8H	-	-	PT2	PS	PT1	PX1	PT0	PX0	xx000000B
SADEN	Slave address mask	B9H									00000000B
WDT_CONTR	Watch Dog Timer Control Register	C1H	WDT_FL AG		EN_WDT	CLR_WD T	IDLE_ WDT	PS2	PS1	PS0	xx000000B
IAP_DATA	ISP/IAP Flash Data Register	C2H									11111111B
IAP_ADDR H	ISP/IAP Address High	C3H									00000000B
IAP_ADDRL	ISP/IAP Address Low	C4H									00000000B
IAP_CMD	ISP/IAP Command Register	C5H							MS1	MS0	xxxxx000B
IAP_TRIG	ISP/IAP_Command Trigger	C6H									xxxxxxxxB
IAP_CONTR	ISP/IAP Control Register	C7H	IAPEN	SWBS	SWRST-	CMD_FAIL -	-	WT2	WT1	WT0	0000x000B
PSW	Program Status Word	D0H	CY	AC	F0	RS1	RS0	OV	-	P	00000000B
ACC	Accumulator	E0H	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000B
B	B Register	F0H									00000000B

Memory Organization



Address Space for STC11Fxx RAM

RAM

There are 256 bytes RAM built in STC11Fxx. The user can visit the leading 128-byte RAM via direct addressing instructions, we name those RAM as *direct RAM* that occupies address space 00h to 7Fh.

Followed 128-byte RAM can be visited via indirect addressing instructions, we name those RAM as *indirect RAM* that occupied address space 80h to *FF*h.

Embedded Flash

There is totally 6K(max) byte flash embedded in the STC11Fxx.

The user can configure the whole flash to store his application program, or he can configure the flash for both storage of application (AP) program and In-System-Program (ISP) code, even he can configure the flash for storage of AP, ISP, and In-Application-Program (IAP) memory.

If there is requirement from the user's application program to store nonvolatile parameters, the user can allocate part of the embedded flash as IAP memory by Part No..

Power Saving

IDLE Mode

An instruction setting SFR **PCON.0** causes the device go into the idle mode, the internal clock is gated off to the CPU but not to the interrupt, timer, WDT and serial port functions.

There are two ways to terminate the idle. Activation of any enabled interrupt will cause **PCON.0** to be cleared by hardware, terminating the idle mode. The interrupt will be serviced, and following RETI instruction, the next instruction to be executed will be the one following the instruction that puts the device into idle. Another way to wake-up from idle is to pull pin RST high to generate internal hardware reset.

Slow Down Mode

A clock divider(CLKDIV) in the frond end of the device is designed to slow down the operation speed of STC11Fxx, to save the operating power dynamically. Different from the same register in STC11Fxx MCU, the content in SFR **CLK_DIV** is always effective without the need to operate in IDLE mode.

SFR: **CLK_DIV**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
-	**-**	**-**	**-**	**-**	**CLKS2**	**CLKS1**	**CLKS0**

{CKS2, CKS1, CKS0}: Clock selector under idle mode

{0, 0,0} : = (default)
In idle mode, clock is not divided (default state)

{0, 0, 1} : =
In idle mode, clock is divided by 2

{0, 1, 0} : =
In idle mode, clock is divided by 4

{0, 1, 1} : =
In idle mode, clock is divided by 8

{1, 0,0} : =
In idle mode, clock is divided by 16

{1, 0, 1} : =
In idle mode, clock is divided by 32

{1, 1, 0} : =
In idle mode, clock is divided by 64

{1, 1, 1} : =
In idle mode, clock is divided by 128

POWER-DOWN Mode

An instruction setting **PCON.1** causes the device go into the *POWER-DOWN* mode. In the *POWER-DOWN* mode, the on-chip oscillator is stopped. The contents of on-chip RAM and SFRs are maintained.

In the Power Down mode, the on-chip oscillator is stopped. The contents of on-chip RAM and SFRs are maintained. The power-down mode can be woken-up by RESET pin, external interrupt INT0 ~ INT3 and keypad interrupt. When it is woken-up by RESET, the program will execute from the address 0x0000. Be carefully to keep RESET pin active for at least 10ms in order for a stable clock. If it is woken-up from I/O, the CPU will rework through jumping to related interrupt service routine. Before the CPU rework, the clock is blocked and counted until 32768 in order for denouncing the unstable clock. To use I/O wake-up, interrupt-related registers have to be enabled and programmed accurately before entering power-down. **Pay attention to have at least one "NOP" instruction subsequent to the power-down instruction if I/O wake-up is used.**

SFR: **PCON**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
SMOD	**SMOD0**	**-**	**POF**	**GF1**	**GF0**	**PD**	**IDL**

SMOD:= Double baud rate of UART interface

0: = (default)

Keep normal baud rate when the UART is used in mode 1,2 or 3.

1 : =

Double baud rate when the UART is used in mode 1,2 or 3.

SMOD0:=

SM0/FE bit select for SFR **SCON.7** ; Setting this bit will set SFR **SCON.7** as Frame Error function. Clearing it to set SCON.7 as one bit of UART mode selection bits. (This bit is serial port related, see the further description about the serial port)

POF:= Power-On flag

This bit will be set after the device was powered on.

It must be cleared by the user's software.

GF1, GF0:= General purpose flags

The user can take them as RAM to hold bit variables.

PD:= Power-Down switch

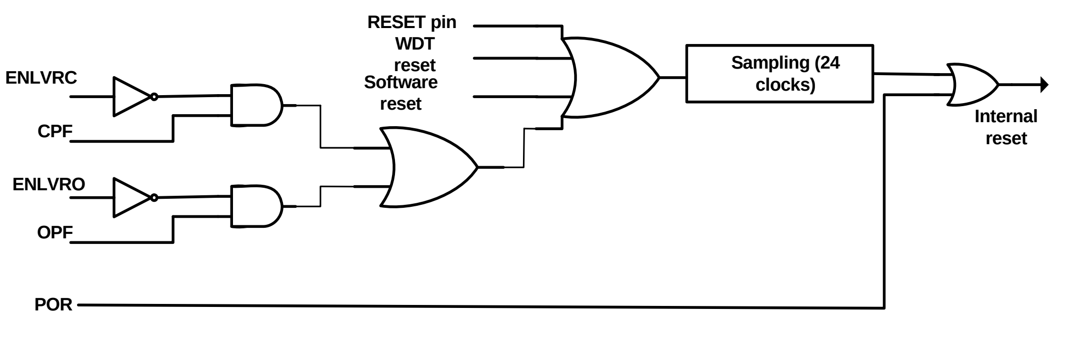
Set this bit to drive the device enter *POWER-DOWN* mode.

IDL:= Idle flag

Set this bit to drive the device enter *IDLE* mode.

Device Resets

There are 6 sources could generate internal reset. They are Power On, RESET pin, Power Monitor Unit(PMU), Watch-Dog-Timer and user-invoked software reset.



STC11Fxx Interrupt Circuit

Reset from RESET Pin

The RESET pin, which is the input to the Schmitt Trigger, is input pin for chip reset. A level change of RESET pin have to keep at least 24 cycles plus 10us in order for CPU internal sampling use.

Watch-Dog-Timer

An overflow of Watch-Dog-Timer will generate a internal reset.

Software RESET

Writing an “1” to **SWRST** bit in SFR **IAP_CONTR** can invoke a internal reset.

Boot Entrance

The following procedure describes how does this device select the boot entrance.

Power-Up

```
If ( (HWBS==0) or (HWBS2==0) ) then
      SWBS = 1
else
      SWBS keeps unchanged
end

If (SWBS==1)    {
      Boot from ISP code
else
      Boot from AP code
```

RESET-pin press

```
If    (HWBS2==0) then
      SWBS = 1
else
      SWBS keeps unchanged
end

If (SWBS==1)    {
      Boot from ISP code
else
      Boot from AP code
```

WDT overflow, Power Monitor Unit reset, and Software reset

```
SWBS keeps unchanged

If (SWBS==1)    {
      Boot from ISP code
else
      Boot from AP code
end
```

Functional Description

I/O Port Configuration

There are 15(max) GPIO available from this device. All IO pins on STC11Fxx may be independently configured to one of four modes: quasi-bidirectional (standard 8051 port output), push-pull output, open-drain output or input-only. All port pins default to quasi-bidirectional after reset. Each port pin has a Schmitt-triggered input for improved input noise rejection. During power-down, all the schmitt-triggered inputs are disabled with the exception of several pins which may be used to wake-up the device. The use can use P3.2(INT0), P3.3(INT1), P3.0(INT), P1.6(INT), to drive this device escape power-down mode. Therefore such kind of pins should not be left floating during power-down.

There are several special function registers designed to configure those I/O ports.

SFR: **P1M0**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
P1M07	**P1M06**	**P1M05**	**P1M04**	**P1M03**	**P1M02**	**P1M01**	**P1M00**

SFR: **P1M1**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
P1M17	**P1M16**	**P1M15**	**P1M14**	**P1M13**	**P1M12**	**P1M11**	**P1M10**

SFR: **P3M0**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
P3M07	**P3M06**	**P3M05**	**P3M04**	**P3M03**	**P3M02**	**P3M01**	**P3M00**

SFR: **P3M1**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
P3M17	**P3M16**	**P3M15**	**P3M14**	**P3M13**	**P3M12**	**P3M11**	**P3M10**

Configuration of I/O port

PxM0*n*	**PxM1*n***	**Port Mode**
0	*0*	Quasi-bidirectional(default)
0	*1*	Push-Pull output
1	*0*	Input Only (High-impedance)
1	*1*	Open-Drain Output

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

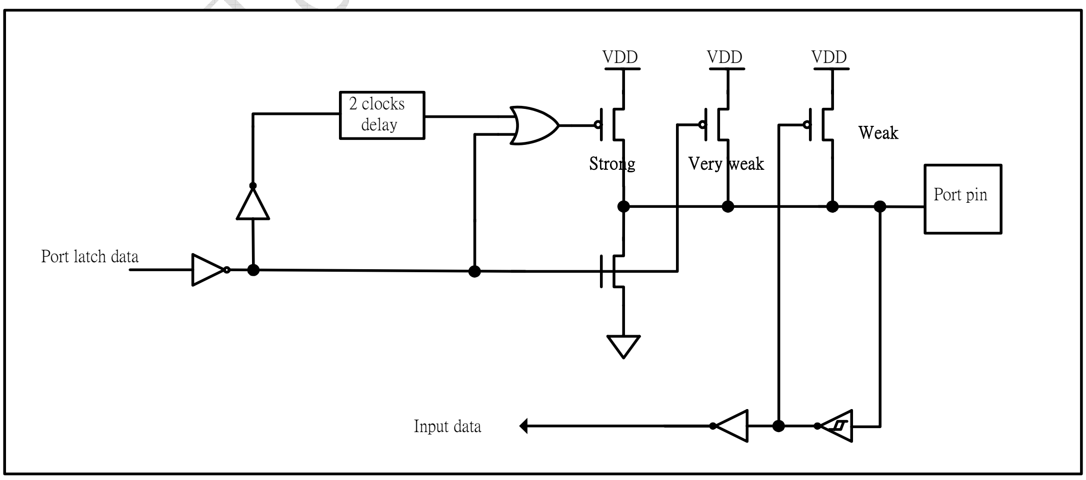
Quasi-bidirectional Mode

Port pins in quasi-bidirectional output mode function similar to the standard 8051 port pins. A quasi-bidirectional port can be used as an input and output without the need to reconfigure the port. This is possible because when the port outputs logic high, it is weakly driven, allowing an external device to pull the pin low. When the pin outputs low, it is driven strongly and able to sink a large current. There are three pull-up transistors in the quasi-bidirectional output that serve different purposes.

One of these pull-ups, called the “very weak” pull-up, is turned on whenever the port register for the pin contains a logic “1”. This very weak pull-up sources a very small current that will pull the pin high if it is left floating.

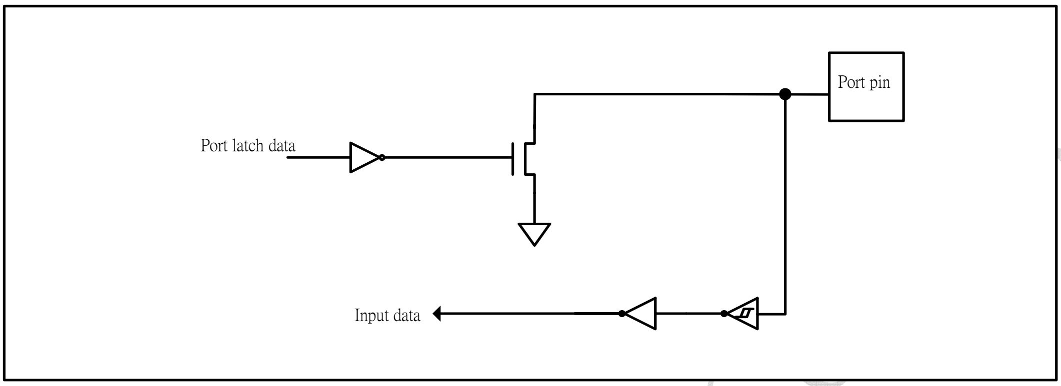
A second pull-up, called the “weak” pull-up, is turned on when the port register for the pin contains a logic “1” and the pin itself is also at a logic “1” level. This pull-up provides the primary source current for a quasi-bidirectional pin that is outputting a ‘1’.   If this pin is pulled low by the external device, this weak pull-up turns off, and only the very weak pull-up remains on. In order to pull the pin low under these conditions, the external device has to sink enough current to over-power the weak pull-up and pull the port pin below its input threshold voltage.

The third pull-up is referred to as the “strong” pull-up. This pull-up is used to speed up low-to-high transitions on a quasi-bidirectional port pin when the port register changes from a logic “0” to a logic “1”. When this occurs, the strong pull-up turns on for two CPU clocks, quickly pulling the port pin high.



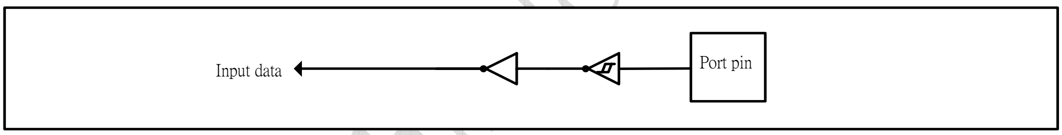
Open-drain Output

The open-drain output configuration turns off all pull-ups and only drives the pull-down transistor of the port pin when the port register contains logic “0”. To use this configuration in application, a port pin must have an external pull-up, typically tied to VDD. The input path of the port pin in this configuration is the same as quasi-bidirection mode.



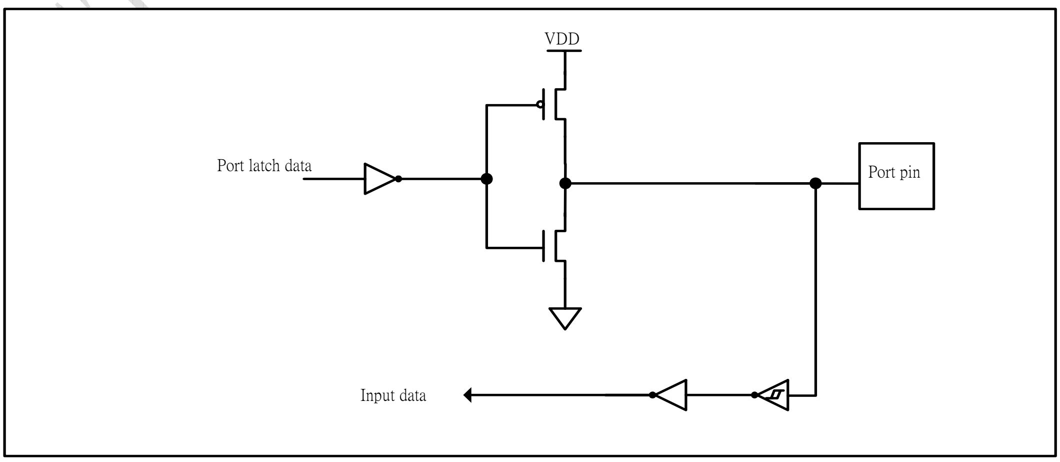
Input-only Mode

The input-only configuration is a Schmitt-triggered input without any pull-up resistors on the pin.



Push-pull Output

The push-pull output configuration has the same pull-down structure as both the open-drain and the quasi-bidirectional output modes, but provides a continuous strong pull-up when the port register contains a logic “1”. The push-pull mode may be used when more source current is needed from a port output.



Timer/Counter

STC11Fxx has two 16-bit timers, and they are named **T0** and **T1**. Each of them can also be used as a general event counter, which counts the transition from 1 to 0.

Since the STC11Fxx is a RISC-like MCU which execute faster than traditional 80C51 MCU from other providers. Based on consideration of compatibility with traditional 80C51 MCUs, the frequency of the clock source for **T0** and **T1** is designed to be selectable between oscillator frequency divided-by-12 (default) or oscillator frequency.

The user can configure T0/T1 to work under mode-0, mode-1, mode-2 and mode-3. It is fully the same to a traditional 80C51 MCU.

There are two SFR designed to configure timers **T0** and **T1**. They are **TMOD**, **TCON**.

The user also should take a glace of SFR **AUXR** which decide the frequency of the clock source driving the **T0** and **T1**.

SFR: **TMOD**(Timer Mode Control Register)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
GATE	**C//T**	**M1**	**M0**	**GATE**	**C//T**	**M1**	**M0**

(for timer1 use) (for timer0 use)

GATE: = Gating control

***0**:* = (default)
Timer *x* is enabled whenever "TR*x*" control bit is set.

***1**:* =
Timer/Counter *x* is enabled only while "/INT*x*" pin is high and **"**TR*x*" control bit is set.

C//T: = Timer or Counter function selector. *0*: =timer, *1*: =counter

***0**:* = (default)
Configure **T*x*** as Timer use

***1**:* =
Configure **T*x*** as Counter use

{M1, M0}: mode select

{0, 0}: =
Configure **T*x*** as 13-bit timer/counter

{0, 1}: =
Configure **T*x*** as 16-bit timer/counter

{1, 0}: =
Configure **T*x*** as 8-bit timer/counter with automatic reload capability

{1, 1}: =
for **T0,** set **TL0** as 8-bit timer/counter, **TH0** is locked into 8-bit timer
for **T1,** set Timer/Counter1 Stopped

SFR: **TCON**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
TF1	**TR1**	**TF0**	**TR0**	**IE1**	**IT1**	**IE0**	**IT0**

TF1: = Timer1 overflow flag.

This bit is automatically set by hardware on **T1** overflow, and will be automatically cleared by hardware when the processor vectors to the interrupt routine.

TR1: = Timer1 run control bit.

0: = (default)
Stop **T1** counting

1: =
Start **T1** counting

TF0: = Timer0 overflow flag.

This bit is automatically set by hardware on **T0** overflow, and will be automatically cleared by hardware when the processor vectors to the interrupt routine.

TR0: = Timer0 run control bit.

0: = (default)
Stop **T0** counting

1: =
Start **T0** counting

IE1: = External Interrupt-1 flag.

This bit is automatically set by hardware on interrupt from the external interrupt-1, and will be automatically cleared by hardware when the processor vectors to the interrupt routine.

IT1: = Interrupt-1 type control bit.

0: = (default)
Set the interrupt-1 triggered by low duty from pin EX1

1: =
Set the interrupt-1 triggered by negative falling edge from pin EX1

IE0: = External Interrupt-0 flag.

This bit is automatically set by hardware on interrupt from the external interrupt-0, and will be automatically cleared by hardware when the processor vectors to the interrupt routine.

IT0: = Interrupt-0 type control bit.

0: = (default)
Set the interrupt-0 triggered by low duty from pin EX1

1: =
Set the interrupt-0 triggered by negative falling edge from pin EX1

SFR: **AUXR** (Auxiliary Register)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
T0X12	**T1X12**	**UARTM0X6**	**BRTR**		**BRTX12**		**S1BRS**

T0X12: = **T0** clock source selector

0: = (default)

Set the frequency of the clock source for **T0** as the oscillator frequency divided-by-12.

It will compatible to the traditional 80C51 MCU.

1: =

Set the frequency of the clock source for **T0** as the oscillator frequency.

It will drive the **T0** faster than a traditional 80C51 MCU.

T1X12: = **T1** clock source selector

0: = (default)

Set the frequency of the clock source for **T1** as the oscillator frequency divided-by-12.

It will compatible to the traditional 80C51 MCU.

1: =

Set the frequency of the clock source for **T1** as the oscillator frequency.

It will drive the **T1** faster than a traditional 80C51 MCU.

URM0X6: = Baud rate selector of UART while it is working under Mode-0

0: = (default)

Set the baud rate of the UART functional block as oscillator frequency divided-by-12.

It will compatible to the traditional 80C51 MCU.

1: =

Set the baud rate of the UART functional block as oscillator frequency divided-by-2.

It will transmit/receive data faster than a traditional 80C51 MCU.

BRTR: = Setting this bit will enable the baud-rate generator of secondary UART to run

0: = (default)

BRTX12: Set this bit to set the clock source for the UART is Fosc, or clear it to set the clock source for or the UART as Fosc/12.

0: = (default)

S1BRS: = The serial port clock source selector

0: = (default)

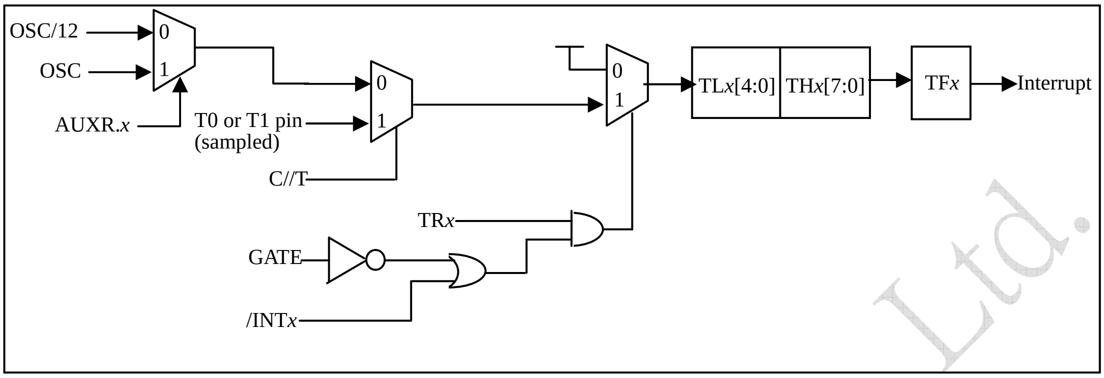
Clear the serial port clock source by **T1** .

1: =

Set the serial port clock source from independence baud-rate generator.

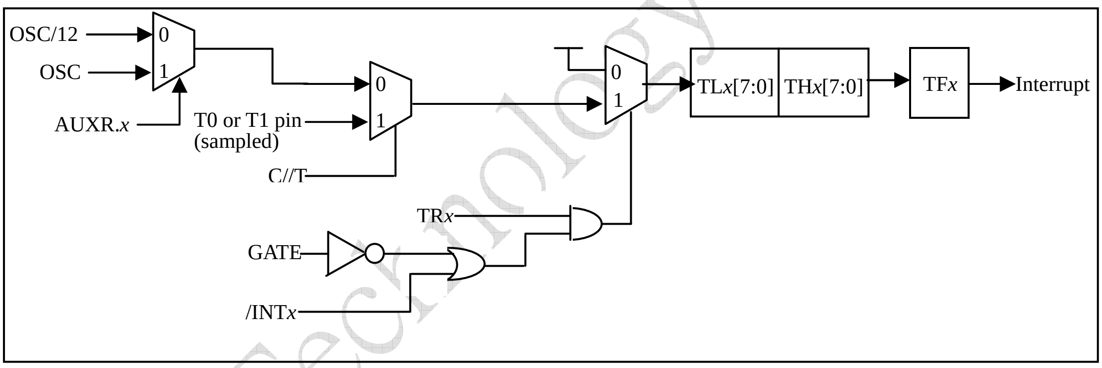
Mode 0

The timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the timer interrupt flag **TFx**. The counted input is enabled to the timer when **TRx** = 1 and either GATE=0 or INTx = 1. Mode 0 operation is the same for Timer0 and Timer1.



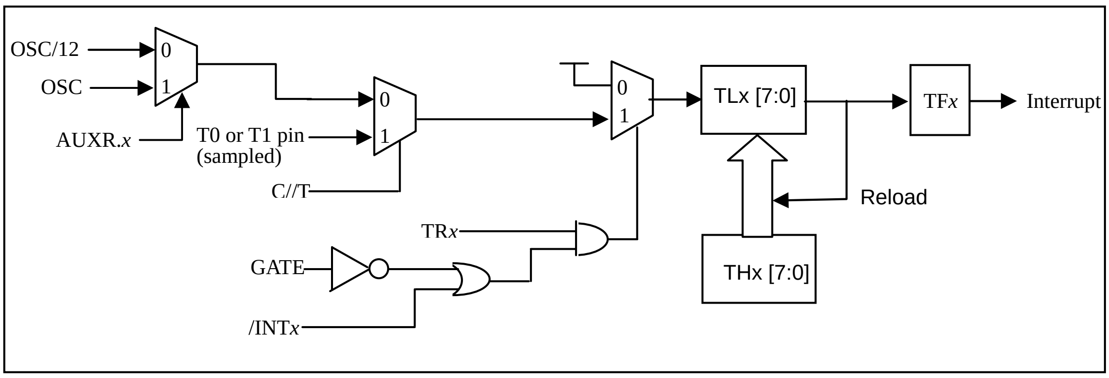
Mode 1

Mode1 is the same as Mode0, except that the timer register is being run with all 16 bits.



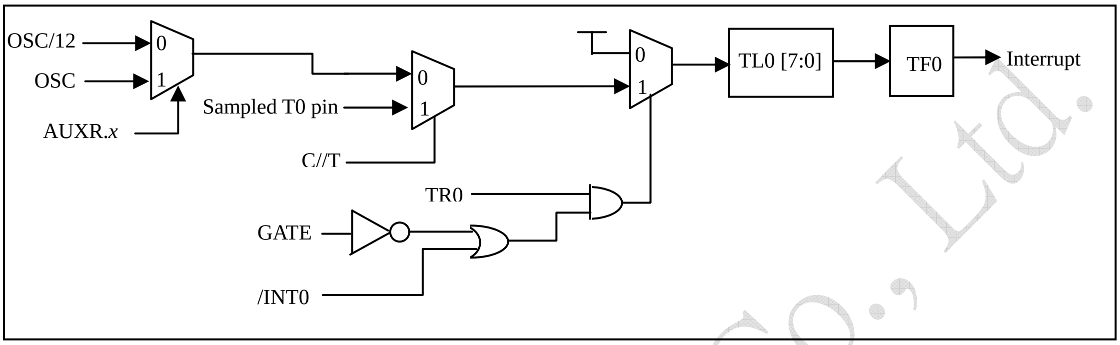
Mode 2

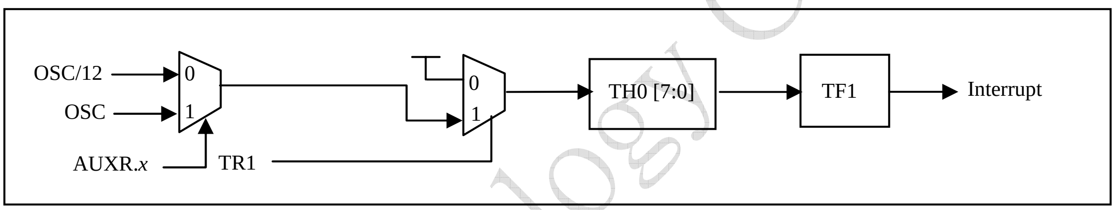
Mode 2 configures the timer register as an 8-bit counter (TL*x*) with automatic reload. Overflow from TL*x* does not only set TF*x*, but also reloads TL*x* with the content of TH*x*, which is determined by user's program. The reload leaves TH*x* unchanged. Mode 2 operation is the same for Timer0 and Timer1.



Mode 3

Timer1 in Mode3 simply holds its count, the effect is the same as setting TR1 = 1. Timer0 in Mode 3 enables TL0 and TH0 as two separate 8-bit counters. TL0 uses the Timer0 control bits such like C/T, GATE, TR0, INT0 and TF0. TH0 is locked into a timer function (can not be external event counter) and take over the use of TR1, TF1 from Timer1. TH0 now controls the Timer1 interrupt.





Interrupt

There are 6 interrupt sources available in STC11Fxx. Each interrupt source can be individually enabled or disabled by setting or clearing a bit in the SFR named **IE**. This register also contains a global disable bit (**EA**), which can be cleared to disable all interrupts at once.

Each interrupt source has two corresponding bits to represent its priority. One is located in SFR named **IPH** and the other in **IP** register. Higher-priority interrupt will be not interrupted by lower-priority interrupt request. If two interrupt requests of different priority levels are received simultaneously, the request of higher priority is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determine which request is serviced. The following table shows the internal polling sequence in the same priority level and the interrupt vector address.

Source	Vector address	Priority within level
External interrupt 0	03H	0 (highest)
Timer 0	0BH	1
External interrupt 1	13H	2
Timer1	1BH	3
Serial Port	23H	4
Low Voltage interrupt	33H	6

The external interrupt /INT0, /INT1 can each be either level-activated or transition-activated, depending on bits **IT0** and **IT1** in register **TCON**. The flags that actually generate these interrupts are bits **IE0** and **IE1** in **TCON**. When an external interrupt is generated, the flag that generated it is cleared by the hardware when the service routine is vectored to *only if the interrupt was transition –activated*, otherwise the external requesting source is what controls the request flag, rather than the on-chip hardware.

The Timer0 and Timer1 interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers in most cases. When a timer interrupt is generated, the flag that generated it is cleared by the on-chip hardware when the service routine is vectored to.

The serial port interrupt is generated by the logical “1” of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. The service routine should poll RI and TI to determine which one to request service and it will be cleared by software.

How does the STC11Fxx take the interrupts

External interrupt pins and other interrupt sources are sampled at rising edge of each clock cycle. The samples are polled during the next clock cycle. If one of the flags was in a set condition of the first cycle, the second cycle of polling cycles will find it and the interrupt system will generate an hardware LCALL to the appropriate service routine as long as it is not blocked by any of the following conditions.

Block conditions

If one of the following conditions happens, a coming interrupt will be blocked.

- An interrupt of equal or higher priority level is already in progress.
- The current cycle(polling cycle) is not the final cycle in the execution of the instruction in progress.
- The instruction in progress is RETI or any write to SFRs **IE**, **IP**, registers.
- The ISP/IAP activity is in progress.

SFR: **IE**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
EA	**-**	**ET2**	**ES**	**ET1**	**EX1**	**ET0**	**EX0**

EA := Global interrupt-controlling register
Set this bit, or any interrupt will be disabled

ET2 := Timer-2 interrupt-controlling register
Setting this bit can enable Timer-2 interrupt

ES := The major UART interrupt-controlling register
Setting this bit can enable major UART interrupt

ET1 := Timer-1 interrupt-controlling register
Setting this bit can enable Timer-1 interrupt

EX1 := INT1 interrupt-controlling register
Setting this bit can enable INT1 interrupt

ET0 := Timer-0 interrupt-controlling register
Setting this bit can enable Timer-0 interrupt

EX0 : INT0 interrupt-controlling register
Setting this bit can enable INT0 interrupt

SFR: ***WAKE_CLK0***

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
-	**RX_PIN_IE**	**T1_PIN_IE**	**T0_PIN_IE**	**-**	**BRTCLKO**	**T1CLKO**	**T0CLKO**

RX_PIN_IE := Wake-Up from RXD Pin
Setting this bit can enable RXD Pin (P3.0 .or. P1.6) wake-up from Power Down Mode.

T1_PIN_IE:= Wake-Up from Timer-1 interrupt
Setting this bit can enable the P3.5 Pin wake-Up from Power Down Mode

T0_PIN_IE:= Wake-Up from Timer-0 interrupt
Setting this bit can enable the P3.5 Pin wake-Up from Power Down Mode.

BRTCLKO:= Setting the bit enable baud-rate generator clock out put to P1.0
“1”: Enable.
“0”:Disable(Default)

T1CLKO:= Setting the bit enable Timer-0 clock output 1/2 time-0 overflow rate to P3.5
““1”: Enable.
“0”:Disable(Default)

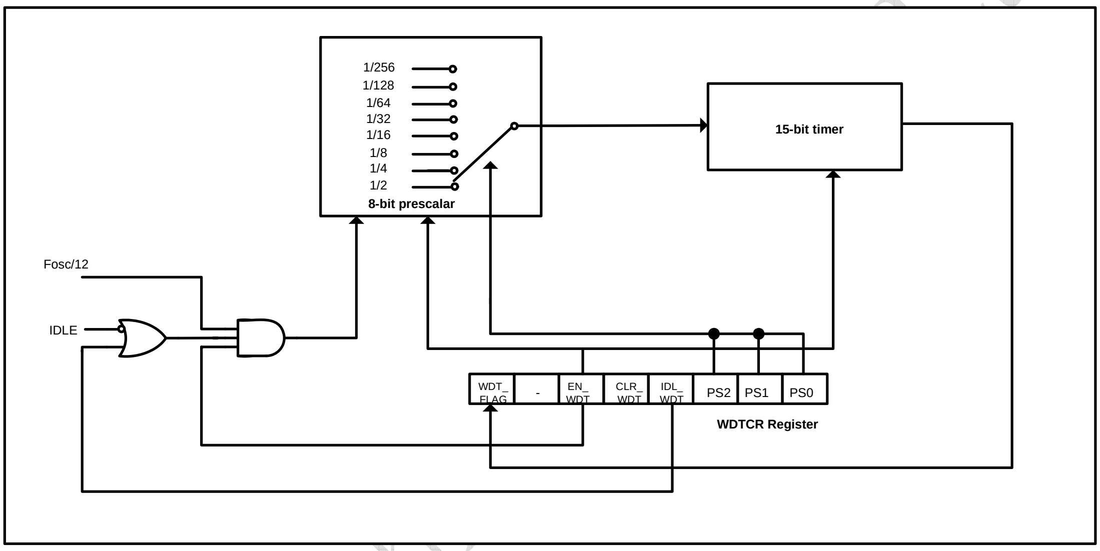
T0CLKO:= Setting the bit enable Timer-1 clock output 1/2 time-1 overflow rate to P3.4
““1”: Enable.
“0”:Disable(Default)

SFR: **IP**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
	PX2	**PT2**	**PS**	**PT1**	**PX1**	**PT0**	**PX0**

Watch Dog Timer

The watch dog timer in STC11Fxx consists of an 8-bit pre-scalar timer and a 15-bit timer. The timer is one-time enabled by setting EN_WDT. Clearing EN_WDT can not stop WDT counting. When the WDT is enabled, software should always reset the timer by writing 1 to CLRW bit before the WDT overflows. If STC11Fxx is out of control by any disturbance, that means the CPU can not run the software normally, then WDT may miss the "writing 1 to CLR_WDT" and overflow will come. WDT overflow reset the CPU to restart.



To make good use of the watch-dog-timer, the user should take notice on SFR **WDT_CONTR**.

SFR: **WDT_CONTR** (WDT Control Register) **C1H**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
WDT_FLAG	**-**	**EN_WDT**	**CLR_WDT**	**IDL_WDT**	**PS2**	**PS1**	**PS0**

WDT_FLAG: = When WDT overflows, this bit is set. It can be cleared by software.

EN_WDT: = Control bit to enable Watch-Dog-Timer. (One-time enabled, can not be disabled)
0: = (default)
Disable Watch Dog Timer
1: =
Enable Watch Dog Timer start counting

CLR_WDT: = Set this bit to recount WDT. Hardware will automatically clear this bit.

IDL_WDT: **=** Behavior controller of the WDT while the device is put under idle

0: = (default)

Stop Watch Dog Timer counting

1: =

Keep Watch Dog Timer counting (so further reset could happen)

{PS2, PS1, PS0}: selector of the WDT pre-scalar output.

{*0, 0, 0*}: = set the pre-scaling value 2
{*0, 0, 1*}: = set the pre-scaling value 4
{*0, 1, 0*}: = set the pre-scaling value 8
{*0, 1, 1*}: = set the pre-scaling value 16
{1*, 0, 0*}: = set the pre-scaling value 32
{*1, 0, 1*}: = set the pre-scaling value 64
{*1, 1, 0*}: = set the pre-scaling value 128
{*1, 1, 1*}: = set the pre-scaling value 256

Universal Asynchronous Serial Port (UART)

The serial port of STC11Fxx is duplex. It can transmit and receive simultaneously. The receiving and transmitting of the serial port share the same SFR **SBUF**, but actually there are two SBUF registers implemented in the chip, one is for transmitting and the other is for receiving. The serial port can be operated in 4 different modes.

Mode 0

Generally, this mode purely is used to extend the I/O features of this device.

Operating under this mode, the device receives the serial data or transmits the serial data via pin RXD, while there is a clock stream shifted via pin TXD which makes convenient for external synchronization. An 8-bit data is serially transmitted/received with LSB first. The baud rate is fixed at 1/12 the oscillator frequency. If **AUXR**.5(**UARTM0X6**) is set, the baud rate is 1/2 oscillator frequency.

Mode1

A 10-bits data is serially transmitted through pin TXD or received through pin RXD. The frame data includes a start bit (*0*), 8 data bits and a stop bit (*1*). After finishing a receiving, the device will keep the stop bit in **RB8** which from SRF **SCON**.

Mode 0

Generally, this mode purely is used to extend the I/O features of this device.

Operating under this mode, the device receives the serial data or transmits the serial data via pin RXD, while there is a clock stream shifted via pin TXD which makes convenient for external synchronization. An 8-bit data is serially transmitted/received with LSB first. The baud rate is fixed at 1/12 the oscillator frequency. If **AUXR**.5 (**URM0X6**) is set, the baud rate is 1/2 oscillator frequency.

Mode1

A 10-bits data is serially transmitted through pin TXD or received through pin RXD. The frame data includes a start bit (*0*), 8 data bits and a stop bit (*1*). After finishing a receiving, the device will keep the stop bit in **RB8** which from SRF **SCON**.

$$\text{Baud Rate (for Mode 1)} = \frac{2^{SMOD}}{32} \times \text{(Timer-1 overflow rate)}$$

Mode2

An 11-bit data is serially transmitted through **TXD** or received through **RXD**. The frame data includes a start bit (*0*), 8 data bits, a programmable 9th bit and a stop bit (1). On transmit; the 9th data bit comes from **TB8** in SFR **SCON**. On receive; the 9th data bit goes into **RB8** in **SCON**. The baud rate is programmable, and permitted to be set either 1/32 or 1/64 the oscillator frequency.

$$\textbf{Baud Rate (for Mode 2)} = \frac{2^{SMOD}}{64} \times \text{Fosc}$$

Mode3

Mode 3 is the same as mode 2 except the baud rate is variable.

$$\textbf{Baud Rate (for Mode 3)} = \frac{2^{SMOD}}{32} \times \text{(Timer-1 overflow rate)}$$

n all four 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*. Reception is initiated in the other modes by the incoming start bit with *1*-to-*0* transition if **REN**=*1*.

There are several SFRs related to serial port configuration described as following.

SFR: **SCON** (Serial Control)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
SM0/FE	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**

FE: = Frame Error bit
This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit.

{SM0, **SM1}: =** Used to set operating mode of the serial port.
- **{0, *0*}: =** set the serial port operate under Mode 0
- **{0, *1*}**: = set the serial port operate under Mode 1
- **{1, *0*}**: = set the serial port operate under Mode 2
- **{1, *1*}**: = set the serial port operate under Mode 3

SM2: = Enable the *automatic address recognition* feature in mode 2 and 3.
If **SM2**=*1*, **RI** will not be set unless the received 9th data bit is 1, indicating an address, and the received byte is a Given or Broadcast address. In mode1, if SM2=1 then RI will not be set unless a valid stop Bit was received, and the received byte is a Given or Broadcast address.

REN: = Enable the serial port reception.

0: = (default)
Disable the serial port reception.

1: =
Enable the serial port reception.

TB8: = The 9th data bit, which will be transmitted in Mode 2 and Mode 3.

RB8: = In mode 2 and 3, the received 9th data bit will be put into this bit.

TI: = Transmitting done flag. After a transmitting has been finished, the hardware will set this bit.

RI: = Receive done flag. After reception has been finished, the hardware will set this bit.

SFR: **SBUF** (Serial Buffer)

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
(data to be transmitted or received data)							

Frame Error Detection

When used for frame error detect, the UART looks for missing stop bits in the communication. A missing bit will set the FE bit in the SCON register. The FE bit shares the SCON.7 bit with SM0 and the function of SCON.7 is determined by PCON.6 (SMOD0). If SMOD0 is set then SCON.7 functions as FE. SCON.7 functions as SM0 when SMOD0 is cleared. When used as FE, SCON.7 can only be cleared by software.

Automatic Address Recognition

There is an extra feature makes the device convenient to act as a master, which communicates to multiple slaves simultaneously. It is really *Automatic Address Recognition*.

There are two SFR **SADDR** and **SADEN** implemented in the device. The user can read or write both of them. Finally, the hardware will make use of these two SFR to "generate" a "compared byte". The formula specifies as following.

Bit[*i*] of **Compared Byte = (SADEN[** *i* **] == 1**)? **SADDR**[*i*] **:** ***x***

For example:

Set **SADDR** = 11000000b
Set **SADEN** = 11111101b
⇨ The achieved "Compared Byte" will be "110000x0" (*x* means don't care)

For another example:

Set **SADDR** = 11100000b
Set **SADEN** = 11111010b
⇨ The achieved "Compared Byte" will be "11100x0x"

After the generic "Compared Byte" has been worked out, the STC11Fxx will make use of this byte to determine how to set the bit **RI** in SFR **SCON**.

Normally, an UART will set bit **RI** whenever it has done a byte reception; but for the UART in the STC11Fxx, if the bit **SM2** is set, it will set **RI** according to the following formula.

RI = (SM2 == *1*) && (SBUF == *Compared Byte*) && (RB8 == *1*)

In other words, not all data reception will respond to RI, while specific data does.

By setting the SADDR and the SADEN, the user can filter out those data byte that he doesn’t like to care. This feature brings great help to reduce software overhead.

The above feature adapts to the serial port when operated in Mode1, Mode2, and Mode3. Dealing with Mode 0, the user can ignore it.

SFR**: BRT**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0

It is used as the reload register for generating the baud-rate of the Major UART

SFR: **AUXR**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
T0X12	**T1X12**	**URM0x6**	**BRTR**		**BRTX12**		**S1BRS**

T0X12 :=

Set this bit to set the clock source for timer 0 is Fosc, or clear it to set the clock source for timer 0 as Fosc/12.

T1X12 :=

Set this bit to set the clock source for timer 0 is Fosc, or clear it to set the clock source for timer 1 as Fosc/12.

URM0x6 :=

Set this bit to set the clock source for the major UART is Fosc/2, or clear it to set the clock source for or the major UART as Fosc/12.

BRTR :=

Setting this bit will enable the baud-rate generator of major UART to run.

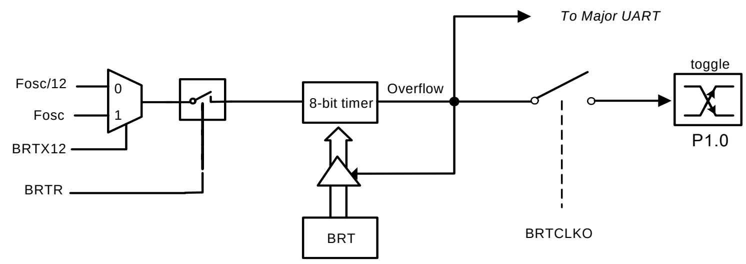
BRTX12:=

Setting this bit can X12 the baud-rate of UART.

S1BRS :=

Setting this bit can enable independence baud-rate generator

Baud-Rate Generator and P1.05 programmable clock output



Baud-Rate Generator for the secondary UART

ST11Fxx is able to generate a programmable clock output on P1.0. When BRTCLKO bit in WAKE_CLKO is set, BRT timer overflow pulse will toggle P1.0 latch to generate a 50% duty clock. The frequency of clock-out is as following :

$$\text{CLKOUT2 (1T Mode)} = \frac{\text{Fosc}/2}{256 - \text{BRT}}$$

$$\text{CLKOUT2 (12T Mode)} = \frac{\text{Fosc}/24}{256 - \text{BRT}}$$

In System Programming and In Application Programming

In System Programming (ISP)

To develop a good program for ISP function, the user has to understand the architecture of the embedded flash.

The embedded flash consists of 128 pages. Each page contains 512 bytes.

Dealing with flash, the user must erase it in page unit before writing (programming) data into it.

Erasing flash means setting the content of that flash as *FF*h. Two erase modes are available in this chip. One is *mass mode* and the other is *page mode*. The *mass mode* gets more performance, but it erases the entire flash. The page mode is something performance less, but it is flexible since it erases flash in page unit.

Unlike RAM's real-time operation, to erase flash or to write (program) flash often takes long time so to wait finish.

Furthermore, it is a quite complex timing procedure to erase/program flash. Fortunately, the STC11Fxx carried with convenient mechanism to help the user read/change the flash content. Just filling the target address and data into several SFR, and triggering the built-in ISP automation, the user can easily erase, read, and program the embedded flash.

There are several SFR designed to help the user implement the ISP functionality.

SFR: **IAP_DATA** *(IAP Flash Data register)* ***0xc2h***

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
Data to be written into flash, or data got from flash							

IAP_DATA is the data port register for ISP/IAP operation. The data in IAP_DATA will be written into the desired address in operating ISP/IAP write and it is the data window of readout in operating ISP read.

SFR: **IAP_ADDRH** *(IAP Flash Address High byte)****0xc3h***

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
Must be cleared to 000			***ISP/IAP address High byte***				

IAP_ADDRH is the high byte address for all ISP/IAP operation.
Against in advertise effect, if one bit of IAP_ADDRH [7:5] is set, the ISP write function must fail.

SFR: **IAP_ADDRL** *(IAP Flash Address Low byte)**0xc4h*

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
ISP/IAP address Low byte							

IAP_ADDRL is the low byte address for all ISP/IAP operation.

SFR: **IAP_CMD** *(ISP Flash-operating Mode Table)**0xc5h***

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
-	-	-	-	-	-	**Mode Selection**	

Mode Selection		To Operate
0	*0*	Standby
0	*1*	AP-memory read
1	*0*	AP-memory/Data-flash program
1	*1*	AP-memory/Data-flash page erase

SFR: ***IAP_TRIG*** *(ISP Sequential Command register to trigger ISP/IAP operation)0xc6h*

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
IAP/SP-Command							

IAP_TRIG is the command port for triggering ISP activity. If IA**SIAP_TRIG** is filled with sequential $5A_H$, $A5_H$ and if **IAP_CONTR**.7 = 1, ISP activity will be triggered.
When this register is read, the device ID of STC11Fxx will be returned (2 bytes). The MSB byte of this device ID is $F2_H$ and LSB byte 02_H. **ISP_ADDRL**.0 is used to select HIGH/LOW byte of the device ID.

SFR: **IAP_CONTR** *(IAP Control register)0xc7h*

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
IAPEN	**SWBS**	**SWRST**	**CMD_FAIL**	**-**	**WAIT**		

IAPEN: = Determine if to Enable ISP/IAP functionality

0: =

Disable ISP program to change flash.

1: =

Enable ISP program to change flash.

SWBS: = Software Boot entrance Selector

0: =

Boot from main-memory.

1: =

Boot from ISP memory.

Note: This bit will be loaded with **HWBS(OR0.3)** *after power-up moment.*

SWRST: = Software Reset trigger

Setting this bit will cause the device reset.

CMD_FAIL: = ISP/IAP Command Fail flag

0: =

The last ISP/IAP command has finished successfully.

1: =

The last ISP/IAP command fails. It could be caused since the access of flash memory was inhibited.

WAIT: = Waiting time selection while the flash is busy.

	CPU Wait time (Oscillator cycle)			
IAP_CONTR[2:0]	Page Erase	Program	Read	Recommended System clock
0 0 0	672384	1760	2	30M~24M
0 0 1	504288	1320	2	24M~20M
0 1 0	420240	1100	2	20M~12M
0 1 1	252144	660	2	12M~6M
1 0 0	126072	330	2	6M~3M
1 0 1	63036	165	2	3M~2M

1 1 0	42024	110	2	2M~1M
1 1 1	21012	55	2	< 1M

Procedures demonstrating ISP function

IAPCMD ← *xxxxx*011$_B$ /* choice page-erasing command */
IAP_CONTR ← 100xx010$_B$ /* set ISPEN=1 to enable flash change. set WAIT=010, 10942 MC; assumed 10M X's*/
IAP_ADDRH ← *(page address high byte)* /* specify the address of the page to be erased */
IAP_ADDRL ← *(page address low byte)*
IAP_TRIG ← 5Ah /* trig IAP activity */
IAP_TRIG ← A5h
(CPU progressing will be hold here)
(CPU continues)

Erase a specific flash page

IAPCMD ← *xxxxx*010$_B$ /* choice byte-programming command */
IAP_CONTR ← 100xx010$_B$ /* set ISPEN=1 to enable flash change. set WAIT=010, 60 MC; assumed 10M X's*/
IAP_ADDRH ← *(Address high byte)* /* specify the address to be programmed */
IAP_ADDRL ← *(Address low byte)*
IAP_DATA ← *(byte date to be written into flash)* /* prepare data source */
IAP_TRIG ← 5Ah /* trig IAP activity */
IAP_TRIG ← A5h
(CPU progressing will be hold here)
(CPU continues)

Program a byte into flash

IAPCMD ← *xxxxx*001$_B$ /* choice byte-read command */
IAP_CONTR ← 100xx010$_B$ /* set IAPEN=1 to enable flash change. set WAIT=010, 11 MC; assumed 10M X's*/
IAP_ADDRH ← *(Address high byte)* /* specify the address to be read */
IAP_ADDRL ← *(Address low byte)*
SCMD ← 5Ah /* trig ISP activity */
SCMD ← A5h
(CPU progressing will be hold here)
(CPU continues and currently **IAP_DATA** contain the desired data byte)

Read a byte from flash

In-Application Program (IAP)

The In-Application Program feature is designed for user to Read/Write nonvolatile *data flash*. It may bring great help to store parameters those should be independent of power-up and power-done action. In other words, the user can store data in *data flash* memory, and after he shutting down the MCU and rebooting the MCU, he can get the original value, which he had stored in.

The user can program the *data flash* according to the same way as ISP program, so he should get deeper understanding related to SFR **IAP_DATA, IAP_ADDRL, IAP_ADDRH, IAP_CMD, IAP_TRIG, and IAP_CONTR**.

The *data flash* can be programmed by the AP program as well as the ISP program.

The ISP program may program the AP memory and *data flash*, while the AP program may program the *data flash* but not the ISP memory. If the AP program desires to change the ISP memory associated with specific address space, the hardware will ignore it.

Other Auxiliary SFRs

SFR: **AUXR1**

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
	UART_P1			**-**	**GF2-**		**DPS-**

UART_P1:=

0: =

Disable

1 : =

Enable

GF2:=

General purpose flag. It can be used us a variable.

DPS:=

The switching bit for access of dual DPTR.

Built-In Oscillator

There is an oscillator built in the STC11Fxx which can be used as the oscillating source replacing the external crystal oscillator in some specific applications.

To enable the built-in oscillator, an user must configure the device by clearing (enable) the bit via a general writer.

Making use of the built-in oscillator saves the cost of a crystal oscillator.

Typically, the frequency of the built-in oscillator is designed as 6MHz at 25℃. Dealing with temperature variation, the frequency could vary from 4.2MHz to 7.8MHz (~30%). It is designed for applications which don’t ask very precise oscillating frequency, but not those applications asking high precision of oscillator frequency.

Instructions Set

DATA TRASFER			
MNEMONIC	DESCRIPTION	BYT	CYC
MOV A, Rn	Move register to Acc	1	1
MOV A, direct	Move direct byte o Acc	2	2
MOV A, @Ri	Move indirect RAM to Acc	1	2
MOV A, #data	Move immediate data to Acc	2	2
MOV Rn, A	Move Acc to register	1	2
MOV Rn, direct	Move direct byte to register	2	4
MOV Rn, #data	Move immediate data to register	2	2
MOV direct, A	Move Acc to direct byte	2	3
MOV direct, Rn	Move register to direct byte	2	3
MOV direct, direct	Move direct byte to direct byte	3	4
MOV direct, @Ri	Move indirect RAM to direct byte	2	4
MOV direct, #data	Move immediate data to direct byte	3	3
MOV @Ri, A	Move Acc to indirect RAM	1	3
MOV @Ri, direct	Move direct byte to indirect RAM	2	3
MOV @Ri, #data	Move immediate data to indirect RAM	2	3
MOV DPTR,#data16	Load DPTR with a 16-bit constant	3	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to Acc	1	4
MOVC A,@A+PC	Move code byte relative to PC to Acc	1	4
MOVX A,@Ri	Move on-chip auxiliary RAM(8-bit address) to Acc	1	3
MOVX A,@DPTR	Move on-chip auxiliary RAM(16-bit address) to Acc	1	3
MOVX @Ri,A	Move Acc to on-chip auxiliary RAM(8-bit address)	1	4
MOVX @DPTR,A	Move Acc to on-chip auxiliary RAM(16-bit address)	1	3
MOVX A,@Ri	Move external RAM(8-bit address) to Acc	1	7
MOVX A,@DPTR	Move external RAM(16-bit address) to Acc	1	7
MOVX @Ri,A	Move Acc to external RAM(8-bit address)	1	7
MOVX @DPTR,A	Move Acc to external RAM(16-bit address)	1	7
PUSH direct	PUSH DIRECT BYTE ONTO STACK	2	4
POP direct	POP DIRECT BYTE FROM STACK	2	3
XCH A, Rn	EXCHANGE REGISTER WITH ACC	1	3
XCH A, direct	EXCHANGE DIRECT BYTE WITH ACC	2	4
XCH A, @Ri	EXCHANGE INDIRECT RAM WITH ACC	1	4
XCHD A, @Ri	EXCHANGE LOW-ORDER DIGIT INDIRECT RAM WITH ACC	1	4

ARITHEMATIC OPERATIONS			
MNEMONIC	DESCRIPTION	BYT	CYC
ADD A, Rn	ADD REGISTER TO ACC	1	2
ADD A, direct	ADD DIRECT BYTE TO ACC	2	3
ADD A, @Ri	ADD INDIRECT RAM TO ACC	1	3
ADD A, #data	ADD IMMEDIATE DATA TO ACC	2	2
ADDC A, Rn	ADD REGISTER TO ACC WITH CARRY	1	2
ADDC A, direct	ADD DIRECT BYTE TO ACC WITH CARRY	2	3
ADDC A, @Ri	ADD INDIRECT RAM TO ACC WITH CARRY	1	3
ADDC A, #data	ADD IMMEDIATE DATA TO ACC WITH CARRY	2	2
SUBB A, Rn	SUBTRACT REGISTER FROM ACC WITH BORROW	1	2
SUBB A, direct	SUBTRACT DIRECT BYTE FROM ACC WITH BORROW	2	3
SUBB A, @Ri	SUBTRACT INDIRECT RAM FROM ACC WITH BORROW	1	3
SUBB A, #data	SUBTRACT IMMEDIATE DATA FROM ACC WITH BORROW	2	2
INC A	INCREMENT ACC	1	2
INC Rn	INCREMENT REGISTER	1	3
INC direct	INCREMENT DIRECT BYTE	2	4
INC @Ri	INCREMENT INDIRECT RAM	1	4
DEC A	DECREMENT ACC	1	2
DEC Rn	DECREMENT REGISTER	1	3

DEC direct	DECREMENT DIRECT BYTE	2	4
DEC @Ri	DECREMENT INDIRECT RAM	1	4
INC DPTR	INCREMENT DPTR	1	1
MUL AB	MULTIPLY A AND B	1	4
DIV AB	DIVIDE A BY B	1	5
DA A	DECIMAL ADJUST ACC	1	4

LOGIC OPERATION			
MNEMONIC	**DESCRIPTION**	**BYT**	**CYC**
ANL A, Rn	AND REGISTER TO ACC	1	2
ANL A, direct	AND DIRECT BYTE TO ACC	2	3
ANL A, @Ri	AND INDIRECT RAM TO ACC	1	3
ANL A, #data	AND IMMEDIATE DATA TO ACC	2	2
ANL direct, A	AND ACC TO DIRECT BYTE	2	4
ANL direct, #data	AND IMMEDIATE DATA TO DIRECT BYTE	3	4
ORL A, Rn	OR REGISTER TO ACC	1	2
ORL A, direct	OR DIRECT BYTE TO ACC	2	3
ORL A, @Ri	OR INDIRECT RAM TO ACC	1	3
ORL A, #data	OR IMMEDIATE DATA TO ACC	2	2
ORL direct, A	OR ACC TO DIRECT BYTE	2	4
ORL direct, #data	OR IMMEDIATE DATA TO DIRECT BYTE	3	4
XRL A, Rn	EXCLUSIVE-OR REGISTER TO ACC	1	2
XRL A, direct	EXCLUSIVE-OR DIRECT BYTE TO ACC	2	3
XRL A, @Ri	EXCLUSIVE-OR INDIRECT RAM TO ACC	1	3
XRL A, #data	EXCLUSIVE-OR IMMEDIATE DATA TO ACC	2	2
XRL direct, A	EXCLUSIVE-OR ACC TO DIRECT BYTE	2	4
XRL direct, #data	EXCLUSIVE-OR IMMEDIATE DATA TO DIRECT BYTE	3	4
CLR A	CLEAR ACC	1	1
CPL A	COMPLEMENT ACC	1	2
RL A	ROTATE ACC LEFT	1	1
RLC A	ROTATE ACC LEFT THROUGH THE CARRY	1	1
RR A	ROTATE ACC RIGHT	1	1
RRC A	ROTATE ACC RIGHT THROUGH THE CARRY	1	1
SWAP A	SWAP NIBBLES WITHIN THE ACC	1	1

BOOLEAN VARIABLE MANIPULATION			
MNEMONIC	**DESCRIPTION**	**BYT**	**CYC**
CLR C	CLEAR CARRY	1	1
CLR bit	CLEAR DIRECT BIT	2	4
SETB C	SET CARRY	1	1
SETB bit	SET DIRECT BIT	2	4
CPL C	COMPLEMENT CARRY	1	1
CPL bit	COMPLEMENT DIRECT BIT	2	4
ANL C, bit	AND DIRECT BIT TO CARRY	2	3
ANL C, /bit	AND COMPLEMENT OF DIRECT BIT TO CARRY	2	3
ORL C, bit	OR DIRECT BIT TO CARRY	2	3
ORL C, /bit	OR COMPLEMENT OF DIRECT BIT TO CARRY	2	3
MOV C, bit	MOVE DIRECT BIT TO CARRY	2	3
MOV bit, C	MOVE CARRY TO DIRECT BIT	2	4

BOOLEAN VARIABLE BRANCH			
MNEMONIC	**DESCRIPTION**	**BYT**	**CYC**
JC rel	JUMP IF CARRY IS SET	2	3
JNC rel	JUMP IF CARRY NOT SET	2	3
JB bit, rel	JUMP IF DIRECT BIT IS SET	3	4
JNB bit, rel	JUMP IF DIRECT BIT NOT SET	3	4
JBC bit, rel	JUMP IF DIRECT BIT IS SET AND THEN CLEAR BIT	3	5

PROAGRAM BRACHING			
MNEMONIC	**DESCRIPTION**	**BYT**	**CYC**
ACALL addr11	ABSOLUTE SUBROUTINE CALL	2	6
LCALL addr16	LONG SUBROUTINE CALL	3	6
RET	RETURN FROM SUBROUTINE	1	4
RETI	RETURN FROM INTERRUPT SUBROUTINE	1	4
AJMP addr11	ABSOLUTE JUMP	2	3
LJMP addr16	LONG JUMP	3	4
SJMP rel	SHORT JUMP	2	3
JMP @A+DPTR	JUMP INDIRECT RELATIVE TO DPTR	1	3
JZ rel	JUMP IF ACC IS ZERO	2	3
JNZ rel	JUMP IF ACC NOT ZERO	2	3
CJNE A, direct, rel	COMPARE DIRECT BYTE TO ACC AND JUMP IF NOT EQUAL	3	5
CJNE A, #data, rel	COMPARE IMMEDIATE DATA TO ACC AND JUMP IF NOT EQUAL	3	4
CJNE Rn, #data, rel	COMPARE IMMEDIATE DATA TO REGISTER AND JUMP IF NOT EQUAL	3	4
CJNE @Ri, #data, rel	COMPARE IMMEDIATE DATA TO INDIRECT RAM AND JUMP IF NOT EQUAL	3	5
DJNZ Rn, rel	DECREMENT REGISTER AND JUMP IF NOT EQUAL	2	4
DJNZ direct, rel	DECREMENT DIRECT BYTE AND JUMP IF NOT EQUAL	3	5
NOP	NO OPERATION	1	1

Absolute Maximum Rating (STC11Fxx)

Parameter	Rating
Operating Voltage	4.5V ~ 5.5V
Operating temperature under bias	-40^{o}C ~ 85^{o}C*1
Storage temperature	-40^{o}C ~ 125^{o}C
Voltage on any pin	-0.5 ~ 5.5V
Operating Frequency	DC ~ 25MHz

*1 Tested by sampling

DC Characteristics (STC11Fxx)

VSS = 0V, TA = 25 ℃, V_{CC} = 5.0V   unless otherwise specified

Symbol	Parameter	Test Condition	Limits			Unit
			min	typ	max	
V_{IH1}	Input High voltage for P1 and P3	Vcc=5.0V	2.0			V
V_{IH2}	Input High voltage for RESET pin	Vcc=5.0V	3.5			V
V_{IL}	Input Low voltage	Vcc=5.0V			0.8	V
I_{OL}	Output Low current	V_{PIN} =0.45V	12	20		mA
I_{OH1}	Output High current(push-pull)	V_{PIN} =2.4V	12	20		mA
I_{OH2}	Output High current(Quasi-bidirectional)	V_{PIN} =2.4V		220		uA
I_{IL1}	Logic 0 input current(Quasi-bidirectional)	V_{PIN} =0.45V		17	50	uA
I_{IL2}	Logic 0 input current(Input-Only)	V_{PIN} =0.45V		0	10	uA
I_{LK}	Input Leakage current(Open-Drain output)	V_{PIN} = V_{CC}		0	10	uA
I_{H2L}	Logic 1 to 0 transition current	V_{PIN} =1.8V		230	500	uA
I_{OP}	Operating current	F_{OSC} = 12MHz		12	30	mA
I_{IDLE}	Idle mode current	F_{OSC} = 12MHz		6	15	mA
I_{PD}	Power down current	V_{CC} =5.0V		0.1	50	uA
R_{RST}	Internal reset pull-down resistance	V_{CC} =5.0V		100		Kohm

Absolute Maximum Rating (STC11LExx)

Parameter	Rating
Operating Voltage	2.4V ~ 3.6V
Operating temperature under bias	-40°C ~ 85°C*1
Storage temperature	-40°C ~ 125°C
Voltage on any pin	-0.5 ~ 3.6V
Operating Frequency	DC ~ 25MHz

*1Tested by sampling

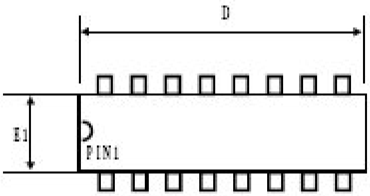
DC Characteristics (STC11LExx)

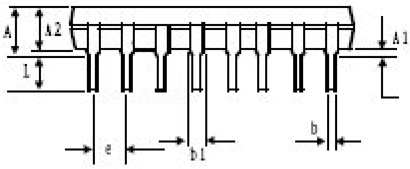
VSS = 0V, TA = 25 ℃, V_{CC} = 3.3V   unless otherwise specified

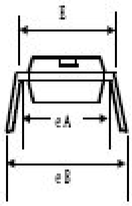
Symbol	Parameter	Test Condition	Limits			Unit
			min	typ	max	
V_{IH1}	Input High voltage for P1 and P3	Vcc=3.3V	2.0			V
V_{IH2}	Input High voltage for RESET pin	Vcc=3.3V	2.8			V
V_{IL}	Input Low voltage	Vcc=3.3V			0.8	V
I_{OL}	Output Low current	V_{PIN} =0.45V	8	14		mA
I_{OH1}	Output High current(push-pull)	V_{PIN} =2.4V	4	8		mA
I_{OH2}	Output High current(Quasi-bidirectional)	V_{PIN} =2.4V		64		uA
I_{IL1}	Logic 0 input current(Quasi-bidirectional)	V_{PIN} =0.45V		7	50	uA
I_{IL2}	Logic 0 input current(Input-Only)	V_{PIN} =0.45V		0	10	uA
I_{LK}	Input Leakage current(Open-Drain output)	V_{PIN} = V_{CC}		0	10	uA
I_{H2L}	Logic 1 to 0 transition current(P1,3)	V_{PIN} =1.4V		100	600	uA
I_{OP}	Operating current	F_{OSC} = 12MHz		9	15	mA
I_{IDLE}	Idle mode current	F_{OSC} = 12MHz		3.5	6	mA
I_{PD}	Power down current	V_{CC} =3.3V		0.1	50	uA
R_{RST}	Internal reset pull-down resistance	V_{CC} =3.3V		100		Kohm

Package Dimension

PDIP-16

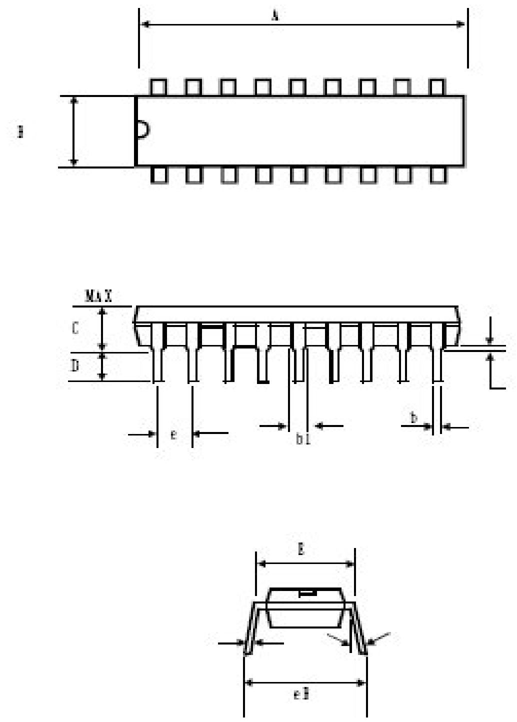






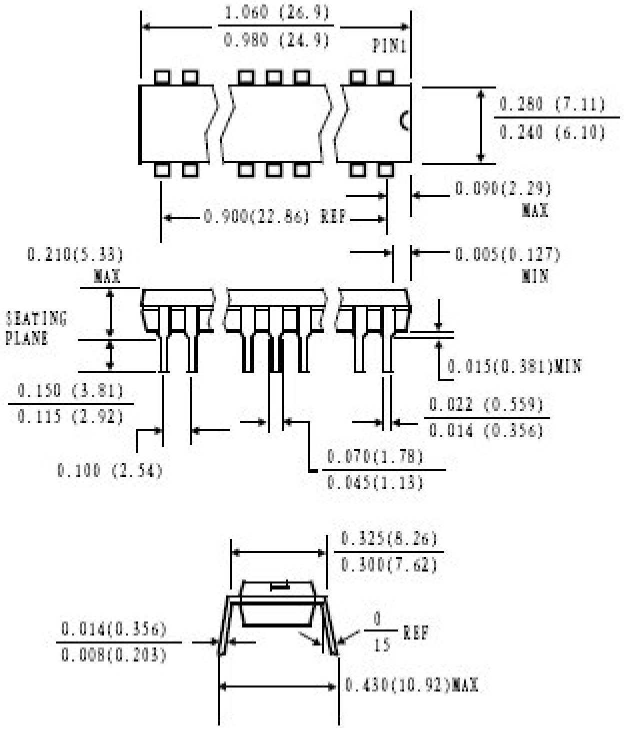
COMMON DIMENSIONS			
(UNITS OF MEASURE = MILLIMETER)			
SYMBOL	MIN	NOM	MAX
A	–	–	4.80
A1	0.50	–	–
A2	3.10	3.30	3.50
b	0.38	–	0.55
b1	0.38	0.46	0.51
D	18.95	19.05	19.15
E	7.62	7.87	8.25
E1	6.25	6.35	6.45
e	2.54BSC		
eA	7.62BSC		
eB	7.62	8.80	10.90
L	2.92	3.30	3.81

PDIP-18

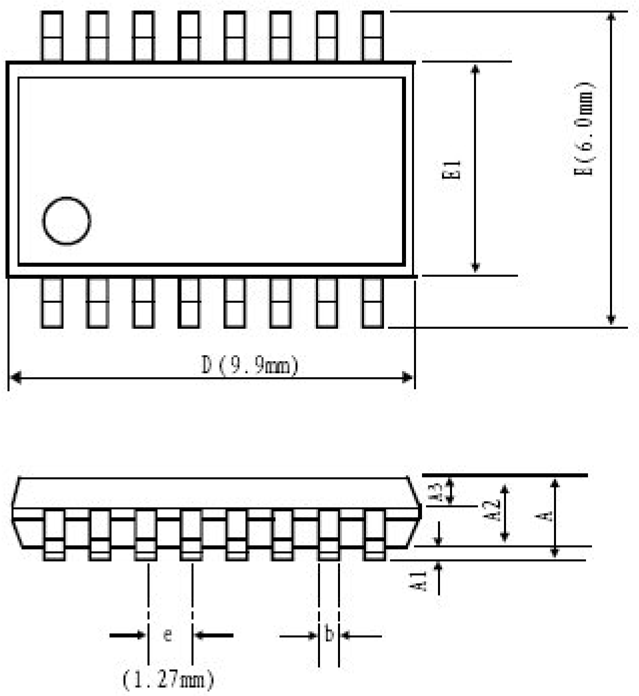


COMMON DIMENSIONS			
(UNITS OF MEASURE = MILLIMETER)			
SYMBOL	MIN	NOM	MAX
A	22.72	–	23.23
B	6.10	–	6.60
C	3.18	–	3.43
D	3.18	–	3.69
e	–	2.54	–
b	0.41	–	0.51
b1	1.27	–	1.78
E	7.49	–	8.00
eB	8.51	–	9.52

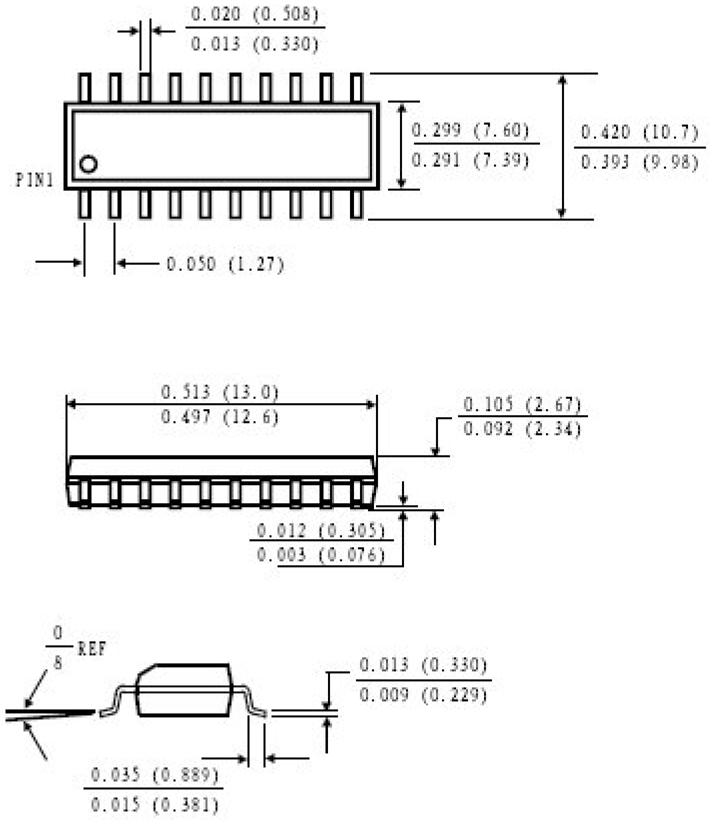
PDIP-18



SOP-16



SOP-20



0.020 (0.508)
0.013 (0.330)
0.299 (7.60)
0.291 (7.39)
0.420 (10.7)
0.393 (9.98)
PIN1
0.050 (1.27)
0.513 (13.0)
0.497 (12.6)
0.105 (2.67)
0.092 (2.34)
0.012 (0.305)
0.003 (0.076)
0
8
REF
0.013 (0.330)
0.009 (0.229)
0.035 (0.889)
0.015 (0.381)

Version History

Version	Date	Page	Description
A1	2008/09		Initial issue
			-
			-
			-