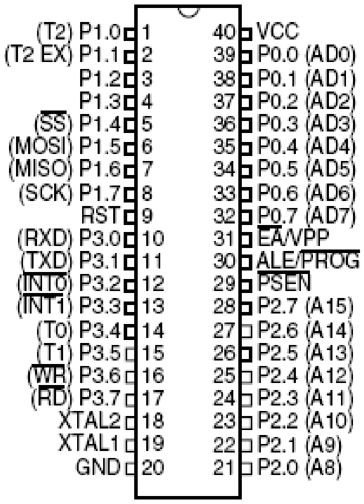
Materiale samlet om uC'en

AT89S8253

AT89S8253 afløser AT89S8252 og AT89S53

Datablad: http://www.atmel.com/dyn/resources/prod_documents/doc3286.pdf



Ale / Prog leave open.

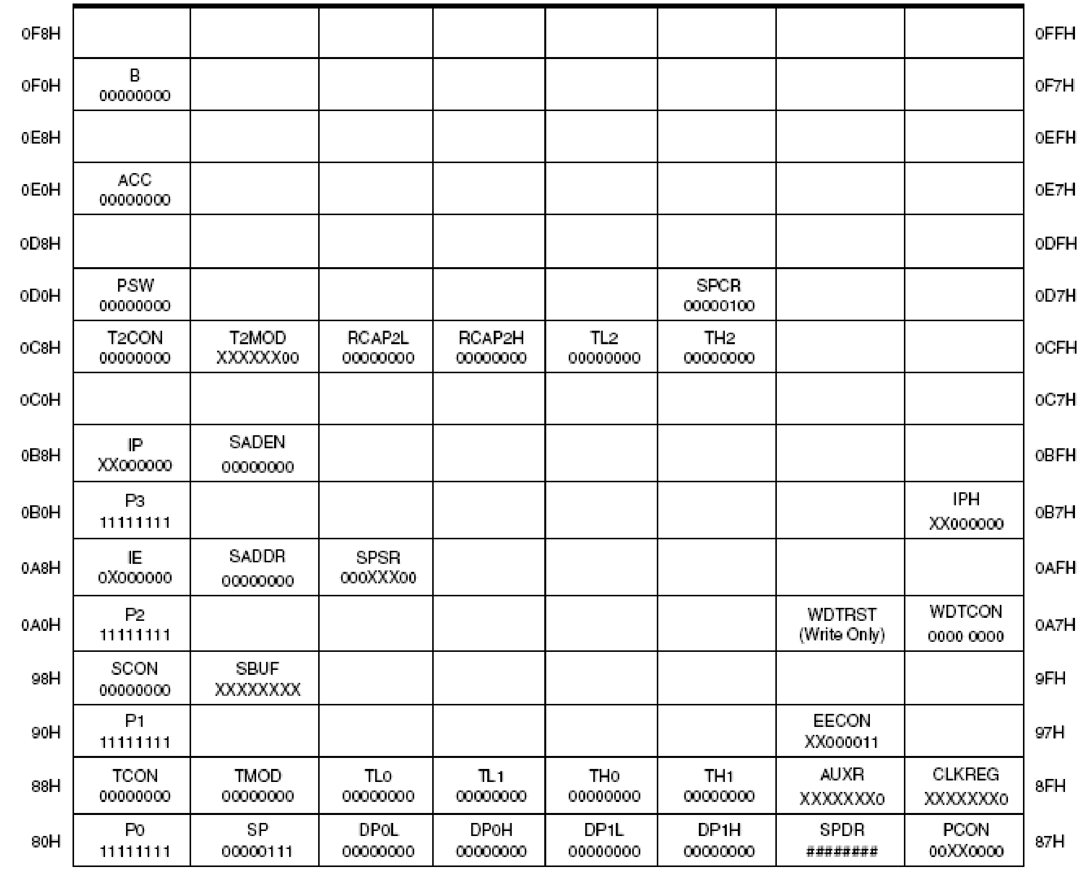
/EA / VPP = 5 Volt ??

/EA/VPP skal have +5 Volt.

SFR 's

0F8H									0FFH
0F0H	B								0F7H
0E8H									0EFH
0E0H	ACC								0E7H
0D8H									0DFH
0D0H	PSW					SPCR			0D7H
0C8H	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2			0CFH
0C0H									0C7H
0B8H	IP	SADEN							0BFH
0B0H	P3							IPH	0B7H
0A8H	IE	SADDR	SPSR						0AFH
0A0H	P2						WDTRST	WDTCON	0A7H
98H	SCON	SBUF							9FH
90H	P1						EECON		97H
88H	TCON	TMOD	TL0	TL1	TH0	TH1	AUXR	CLKREG	8FH
80H	P0	SP	DP0L	DP0H	DP1L	DP1H	SPDR	PCON	87H

Table 5-1. AT89S8253 SFR Map and Reset Values



0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000					SPCR 00000100			0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000	SADEN 00000000							0BFH
0B0H	P3 11111111							IPH XX000000	0B7H
0A8H	IE 0X000000	SADDR 00000000	SPSR 000XXX00						0AFH
0A0H	P2 11111111						WDTRST (Write Only)	WDTCON 0000 0000	0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111						EECON XX000011		97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXXXXXX0	CLKREG XXXXXXX0	8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR ########	PCON 00XX0000	87H

Note: # means: 0 after cold reset and unchanged after warm reset.

SPI

SPI interface'en, Serial Peripheral Interface, SPI, tillader højspeed synkron datatransfer mellem AT89S8253 og tilsluttede enheder, eller mellem 2 processorer.

SPCR, SPI Control Register
Reset værdi = 0000 0100B

7	6	5	4	3	2	1	0
SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
SPI Interrupt enable	SPI Enable	1=> LSB først, 0=> MSB først	1=> Master 0=> Slave	Clock polaritet	Clock fase	00 => f/4 01 => f/16 10 => f/64 11 => f/128	

SPSR, SPI Status register
Reset værdi = 000xxx00B

7	6	5	4	3	2	1	0
SPIF	WCOL	LDEN	-	-	-	DISSO	ENH
SP Interrupt flag. Sættes, når en sending er færdig	Write collision flag					Disable slave output bit	Enhanced SPI Mode

SPI data register: SPDR, Værdier i SPDR ændres ikke ved reset.

/SS	p1.4	/(Slave Select) Bruges, hvis uC fungerer som slave !
MOSI	P1.5	Master Out, Slave In. Forbindes til slavens MOSI
MISO	P1.6,	Master In, Slave Out, Forbindes til slavens MISO
SCK	P1.7	Shift Clock,

Efter opsætning, skrives en byte til dataregisteret, startes SPI clocken, og data sendes så automatisk ud til slaven på MOSI.

Når 1 byte er sendt, stopper clocken, og bit SPIF flag sættes. (End of transmission flag) - ?? Og modtaget byte flyttes til Read Bufferen. SPDR- registeret er både en læse og en skrive, altså 2 fysiske forskellige registre.

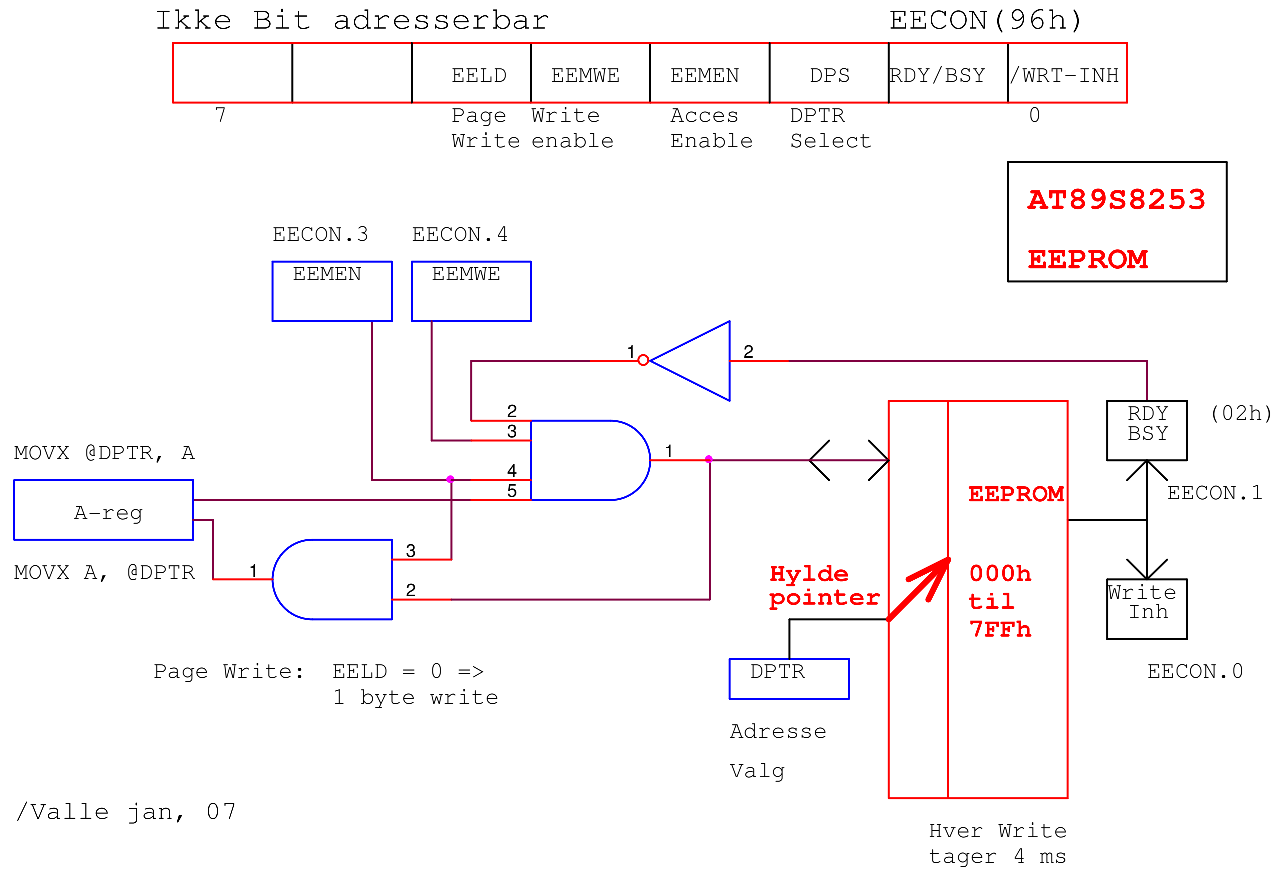
Hvis både SPI-interrupt bit SPIE og den serielle port interrupt bit ES er sat, udløses et interrupt.

Først skal alle øvrige bit opsættes, og til sidst SPE

```
            ORL SPCR, #10h              ;
            ORL SPCR, #83h              ; Interrupt enable og baudrate = f/128
                                        ; eller blot mov SPCR, #93h
            ORL SPCR, #40h              ; start SPI

            SETB EA                     ; enable interrupt i det hele taget
            SETB ES                     ; Enable seriel interrupt. ( SPI interrupt er or'et
                                        ; med UART )

; send data

            Mov SPDR, #3Ah              ; 3A sendes
Send1:
            Mov acc, SPSR ; læs statusregister

            JNB ACC.7, Send1            ; vent på, der er sendt
```

2 K EEPROM

Se evt. i primer ang. Skrivning til, og læsning fra EEPROM



EEPROM kan præ-loades med data i KEIL:

```
ORG 3000h
DB #3Ah
DB #46h
Osv.
```

Eksempel på Læsning fra EEPROM

```
ORL EECON, #EEMEN    ; Enable EEPROMREAD
Mov DPTR, #0h        ; Læseadresse
Movx a, @dptr        ; Læs
mov Ft1, A           ; Flyt
Inc DPTR
Movx a, @dptr        ; Læs
mov Ft10, A          ; Flyt
Inc DPTR
Movx a, @dptr        ; Læs
mov Ft100, A         ; Flyt
```

<u>Eksempel på skrivning til EEPROM:</u>

```
              ORL EECON,#EEMEN      ;enable EEPROM accesses
              ORL EECON,#EEMWE      ;enable EEPROM writes
              ORL EECON,#EELD       ;enable EEPROM page load
              Mov DPTR, #0h         ; load datapointer EEPROMADresse
              MOV A,ft1             ;byte loaded into Accumulator
              MOVX @DPTR,A          ;Load byte to data buffer
              INC DPTR              ;increment DPTR(EEPROM address)
              MOV A,ft10            ;byte loaded into Accumulator
              MOVX @DPTR,A          ;Load byte to data buffer
              INC DPTR              ;increment DPTR(EEPROM address)

;LOAD LAST BYTE AND INITIATE PAGE WRITE
              XRL EECON,#EELD       ;næste MOVX vil starte write cyclus
              MOV A,ft100           ;byte loaded into Accumulator
              MOVX @DPTR,A          ;Load byte to data buffer
              mov EECON, #0h        ; disable eeprom
```

```
;****************************************
```

<u>EEPROM:</u>

Der er indbygget x Kbyte EEPROM, fra adresse 000h til 7ffh

ON-Chip EEPROM vælges ved at sætte bit EEMEN i EECON registeret på adr 96h. (Internal EEPROM access enable.)
EEPROM adresse er fra 000h til 7ffh. Der skal bruges en MOVX instruktion sammen med en data-pointer.

Bit EEMWE i EECON (EEPROM data memory write enable bit.)skal sættes før der kan skrives til EEPROM'en. Bør resettes efter write, for at beskytte EEPROM-indholdet.

Progress kan iagttages ved at læse bit RDY//BSY i EECON reg. (RDY/BSY (Ready/Busy) flag for the data EEPROM memory.) Er skrivecyklus færdig, er bittet sat !!

Hver skrivning til EEPROM tager 4 ms. Men der er mulighed for at skrive op til 32 byte page ad gangen.
Hvis bit EELD i EECON er sat, (EEPROM data memory load enable bit.) vil bytes skrevet til EEPROM'en kun være i en buffer. Før sidste MOVX skal EELD cleares, hvorefter alle (32) bytes skrives til EEPROM'en med sidste byte, og det tager kun 4 ms.

Brug evt en bestemt datapointer til EEprom skrivning.

EECON, Adr 96h

			EELD	EEMWE	EEMEN	DPS	RDY- /BSY	/WRT- INH
			EEPROM Data memory load enable bit	EEPROM data memory write enable bit	Internal EEPROM acces enable	Data pointer register select	Ready-/busy flag for EEPROM Memory	
Bit	7	6	5	4	3	2	1	0

Below is sample code that writes one page of data to EEPROM:

```
EEMWE EQU 00010000B                 ;EEPROM data memory write enable
EEMEN EQU 00001000B                 ;internal EEPROM access enable
EELD EQU 00100000B                  ;EEPROM memory load enable bit
ORL EECON,#EEMEN                    ;enable EEPROM accesses
ORL EECON,#EEMWE                    ;enable EEPROM writes
ORL EECON,#EELD                     ;enable EEPROM page load

Mov DPTR, #xxxh                     ; load datapointer
MOV R7,#1FH                         ;0x1FH = 31 PAGELOAD:
MOV A,#055H                         ;byte loaded into Accumulator
MOVX @DPTR,A                        ;Load byte to data buffer
INC DPTR                            ;increment DPTR(EEPROM address)
DJNZ R7,PAGELOAD                    ;check if 31 bytes have been loaded

;LOAD LAST BYTE AND INITIATE PAGE WRITE

MOV A,#055H                         ;byte loaded into Accumulator
XRL EECON,#EELD                     ;the next will MOVX will start the write cycle
MOVX @DPTR,A                        ;Load byte to data buffer
```

At higher frequencies, if polling the RDY/BSY bit during AT89S8253 EEPROM writes, it becomes necessary to poll for both the start and the end of the EEPROM write cycle. The sample code below illustrates this:

```
WRITESTART:              ;Wait for write cycle to start (RDY/BSY=low)

MOV VAR,096H             ;096H is the EECON SFR
JB VAR.1,Writestart
WAIT:                    ;Wait until write cycle is finished
MOV VAR,096H
JNB VAR.1,Wait
```

EEPROM Read/Write Examples

```
; The EECON register is not bit-addressable, so Boolean operations
are used to control functions and test bits.

EECON DATA 96H             ; watchdog and memory control register
EEMEN EQU 00001000B        ; EEPROM access enable bit
EEMWE EQU 00010000B        ; EEPROM write enable bit
```

```
EELD EQU 00100000B          ; EEPROM page load enable bit
WRTINH EQU 00000001B        ; EEPROM WRTINH bit
RDY EQU 00000010B           ; EEPROM RDY/BSY bit

         ; EEPROM read example.

orl EECON, #EEMEN           ; enable EEPROM accesses
mov dptr, #ADDRESS          ; address to read
movx a, @dptr               ; read EEPROM
xrl EECON, #EEMEN           ; disable EEPROM accesses


&

EEPROM byte write example, utilizing fixed delay for write cycle.
Delay is worst case (10 ms). Code for delay is not shown.
Write is preceded by a write inhibit check, but code to handle an
inhibit condition is not shown. Write is followed by verify
(read and compare), but code to handle verification failure
is not shown.

orl EECON, #EEMEN           ; enable EEPROM accesses
orl EECON, #EEMWE           ; enable EEPROM writes
mov a, EECON                ; get EEPROM status
anl a, #WRTINH              ; check WRTINH
jz ERROR                    ; jump if inhibited
mov dptr, #ADDRESS          ; address to write
mov a, #DATA                ; data to write
movx @dptr, a               ; write EEPROM
call DELAY_10_MS            ; wait 10 ms
movx a, @dptr               ; read EEPROM
cjne a, #DATA, ERROR        ; jump if data compare fails
xrl EECON, #EEMWE           ; disable EEPROM writes
xrl EECON, #EEMEN           ; disable EEPROM accesses
```

256 byte intern RAM

Der er 2 ”banke” af RAM. De laveste 128 byte, til 7Fh, kan skrives direkte. Skrives direkte til en adresse over 7Fh, menes der en SFR-register. Men parallel hermed på de samme adresser er der 128 Byte RAM. De skal blot skrives til indirekte.

Mov R0, #80h
Mov @R0, #data

Stakken kan uden problemer sættes herop ! Den foregår ved indirekte adressering !!

32 I/O pins

Port0,
Har ikke interne Pull Ups. Open Drain
Kan synke 6 ttl-inputs.

Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink six TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Ingen interne pull ups

Consequently P0 lines that are being used as output port lines are open drain. Writing a 1 to the bit latch leaves both output FETs off, so the pin floats. In that conditions it can be used as a high-impedance input.

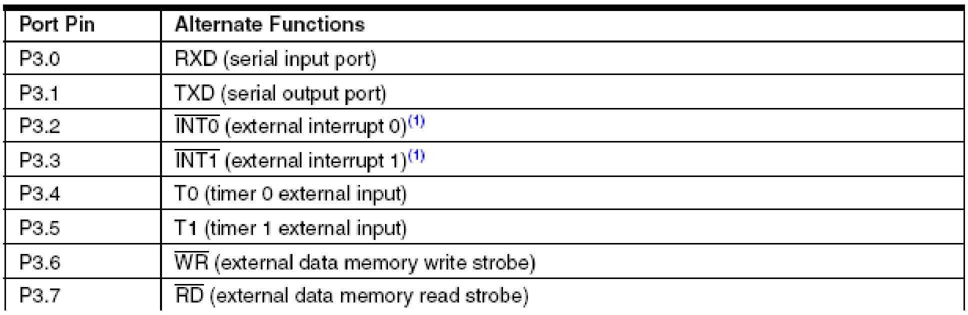
Port 1 is an 8-bit bi-directional I/O port with internal pull-ups. Iout low ~ 150 uA.The Port 1 output buffers can sink/source six TTL inputs.
Som 89C4051. Interne pullups.

Port2:
8 bit bidirektionel, med interne pul lups. Iout low ~ 150 uA.

Port 2 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 2 output buffers can sink/source six TTL inputs. Interne pul lups.

Port3:
8 bit bidirektionel, med interne pul lups. Iout low ~ 150 uA.

Port 3 is an 8-bit bi-directional I/O port with internal pull-ups. The Port 3 output buffers can sink/source six TTL inputs.



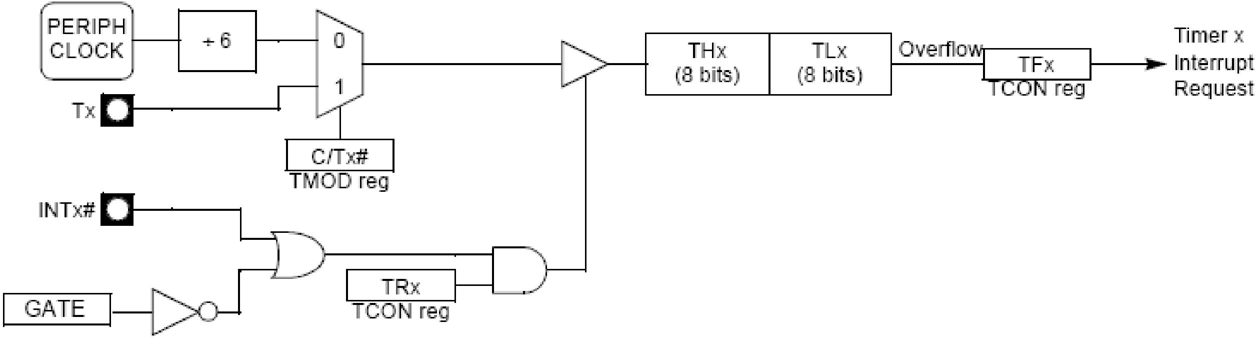
Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)[1]
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)[1]
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

3 stk 16 bit tællere

Hardware description side 81

Timer 0 og timer 1 virker som i AT89C4051

Timer/Counter x (x = 0 or 1) in Mode 1

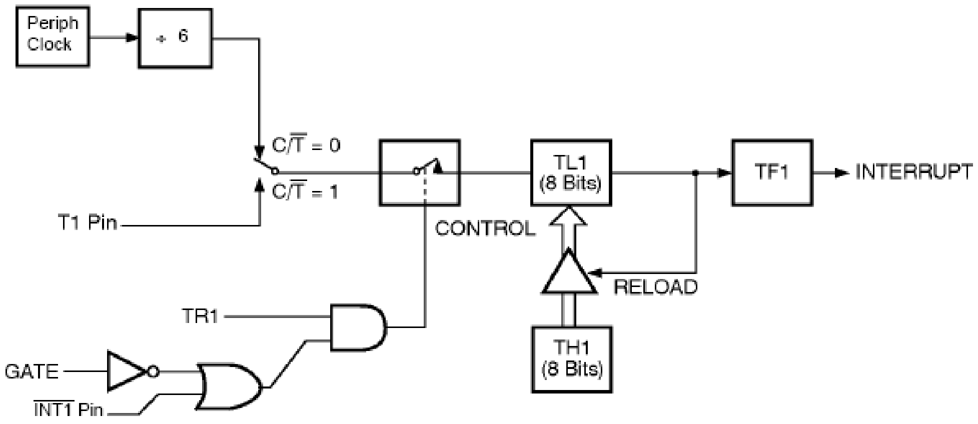


Timer 1

Mode 2 (8-bit Timer with Auto Reload)

Mode 2 configures Timer 1 as an 8-bit timer (TL1 register) with automatic reload from the TH1 register on overflow (see Figure 2-11). TL1 overflow sets the TF1 flag in the TCON register and reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged.

Figure 2-14. Timer/Counter 1 Mode 2: 8-bit Auto-reload



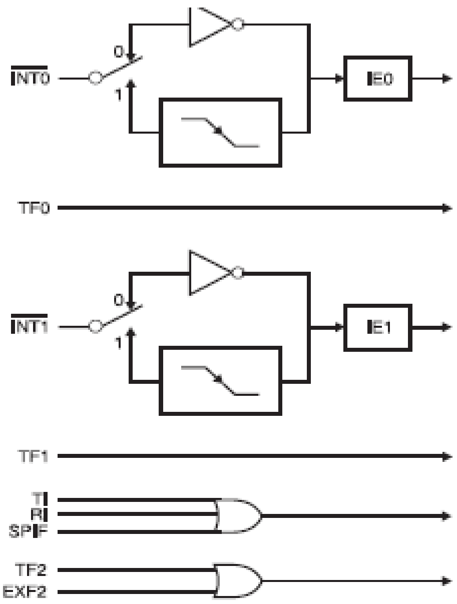
Timer 2 kan bruges til baudrate generering samtidig med, at den kan bruges som ekstern interrupt..

9 interrupts

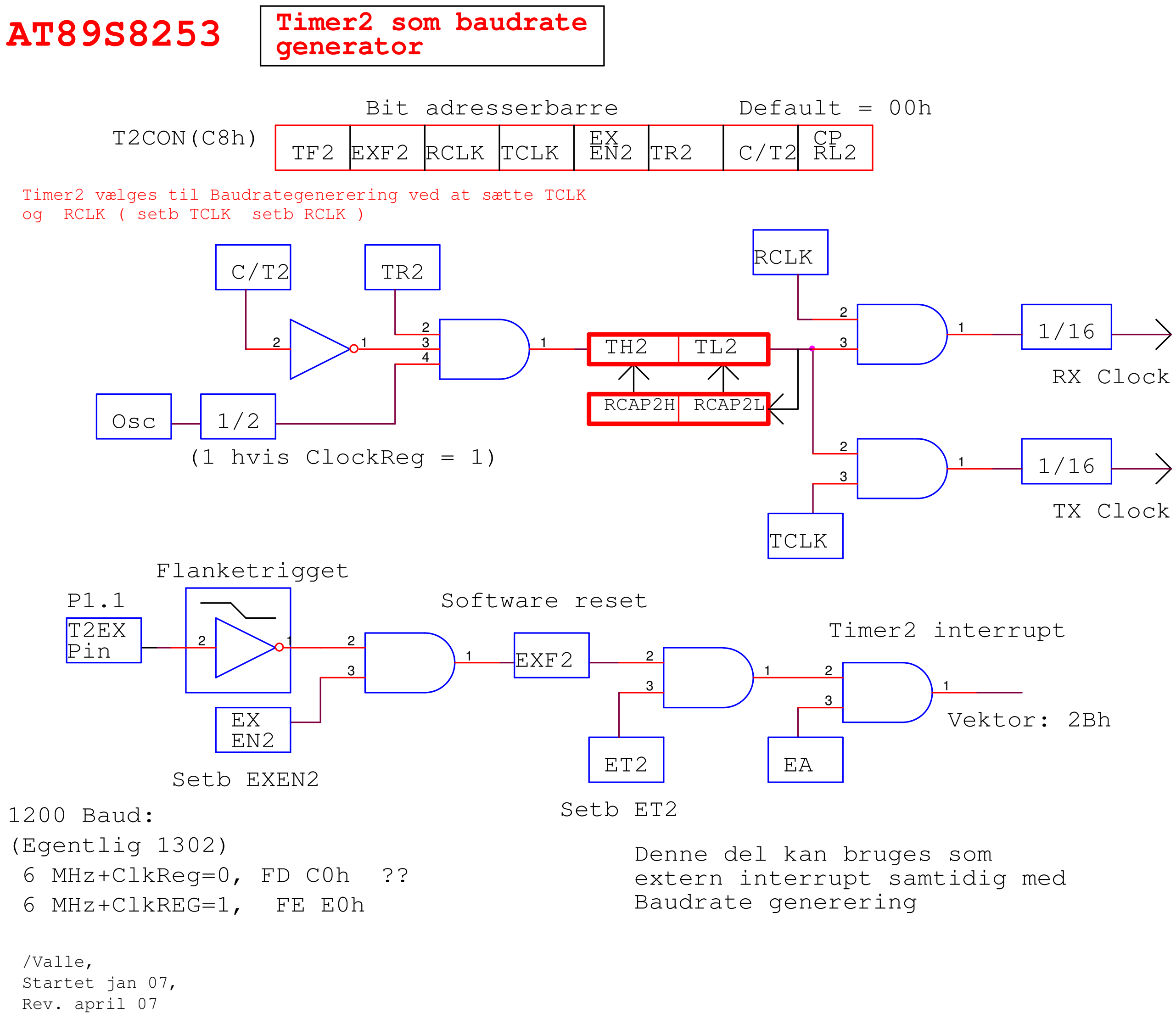
Table 2-27. Interrupt Priority Level

	Source	Priority Within Level
1	IE0	(highest)
2	TF0	
3	IE1	
4	TF1	
5	RI + TI	
6	TF2 + EXF2	(lowest)

Interrupt	Source	Vector Address
System Reset	RST or POR or BOD	0000H
External Interrupt 0	IE0	0003H
Timer 0 Overflow	TF0	000BH
External Interrupt 1	IE1	0013H
Timer 1 Overflow	TF1	001BH
Serial Port	RI or TI or SPIF	0023H



Udvidet UART i AT89S8253



Timer2 vælges til Baudrategenerering ved at sætte TCLK og RCLK (setb TCLK setb RCLK)

(1 hvis ClockReg = 1)

1200 Baud:
(Egentlig 1302)
6 MHz+ClkReg=0, FD C0h ??
6 MHz+ClkREG=1, FE E0h

Denne del kan bruges som extern interrupt samtidig med Baudrate generering

/Valle,
Startet jan 07,
Rev. april 07

$$Baudrate = \frac{F_{Osc} \cdot (1 + 1 \cdot Clockreg))}{32 \cdot (RCAP2H, RCAP2L)} \quad \text{1200 Baud med 12 MHz krystal er egentlig } \frac{1200 \cdot 12}{11{,}0592} = 1302$$

RCAP2H = FEh, RCAP2L = E0h. (Clockreg=1, Krystal=6MHz, Baudrate=1302)

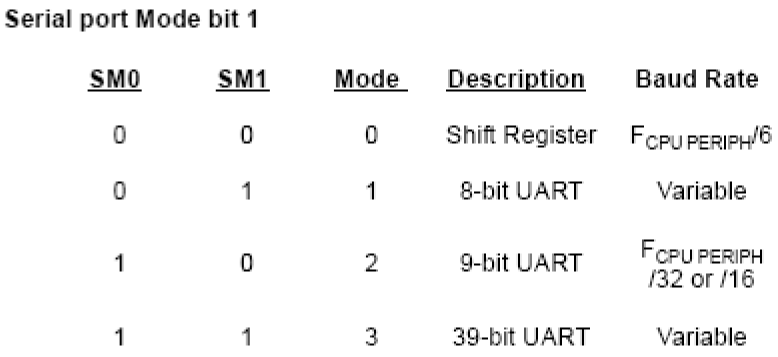
Se: http://www.8052.com/tut8052.phtml
http://www.8052.com/tut8051.phtml

; Opsætning af Timer 2 som baudrate generator

```
        mov TL2, #BaudrateL              ; 1. startværdi sættes
        mov TH2, #BaudrateH
        mov RCAP2L, #BaudrateL           ; genload værdier
```

```
                mov RCAP2H, #BaudrateH
                setb C_T2      ;
                setb RCLK
                SETB TCLK

;Enable Seriel
                setb SM1                                   ;  UART Mode 1, 8 bit asyncron
                SETB REN                                   ; Recieve enable
                Setb TR2                                   ; Sæt Timer 2 igang
                Setb ES                                    ; Enable seriel interrupt
```

Mode 1: 10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

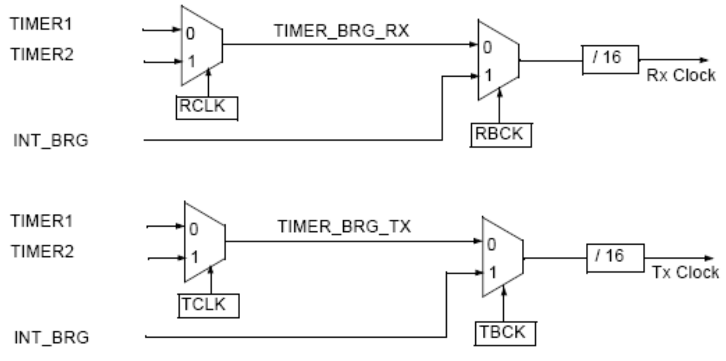


Serial port Mode bit 1

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	$F_{CPU\ PERIPH}/6$
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	$F_{CPU\ PERIPH}$ /32 or /16
1	1	3	39-bit UART	Variable

The Baud Rate Generator for transmit and receive clocks can be selected separately the T2CON and BDRCON registers.

Figure 2-19. Baud Rate Selection



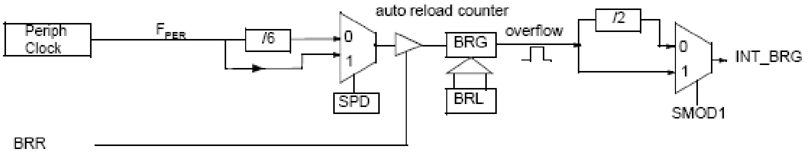
Baudrate selection

TCLK (T2CON)	RCLK (T2CON)	TBCK (BDRCON)	RBCK (BDRCON)	Clock Source UART Tx	Clock Source UART Rx
0	0	0	0	Timer 1	Timer 1
1	0	0	0	Timer 2	Timer 1
0	1	0	0	Timer 1	Timer 2
1	1	0	0	Timer 2	Timer 2
X	0	1	0	INT_BRG	Timer 1
X	1	1	0	INT_BRG	Timer 2
0	X	0	1	Timer 1	INT_BRG
1	X	0	1	Timer 2	INT_BRG
X	X	1	1	INT_BRG	INT_BRG

2.13.3 Internal Baud Rate Generator (BRG)

When the internal Baud Rate Generator is used, the Baud Rates are determined by the BRG overflow depending on the BRL reload value, the value of SPD bit (Speed Mode) in BDRCON register and the value of the SMOD1 bit in PCON register.

Figure 2-20. Internal Baud Rate



- The baud rate for UART is token by formula:

$$\text{Baud\_Rate} = \frac{2^{SMOD1} \text{ x } F_{PER}}{6^{(1-SPD)} \text{ x } 32 \text{ x } [256 - (BRL)]}$$

$$(BRL) = 256 - \frac{2^{SMOD1} \text{ x } F_{PER}}{6^{(1-SPD)} \text{ x } 32 \text{ x Baud\_Rate}}$$

Table 2-15. Example of computed value when X2=1, SMOD1=1, SPD=1
Example of computed value when X2=1, SMOD1=1, SPD=1

Baud Rates	F_{OSCA} = 16.384 MHz		F_{OSCA} = 24 MHz	
	BRL	Error (%)	BRL	Error (%)
115200	247	1.23	243	0.16
57600	238	1.23	230	0.16
38400	229	1.23	217	0.16
28800	220	1.23	204	0.16
19200	203	0.63	178	0.16
9600	149	0.31	100	0.16
4800	43	1.23	-	-

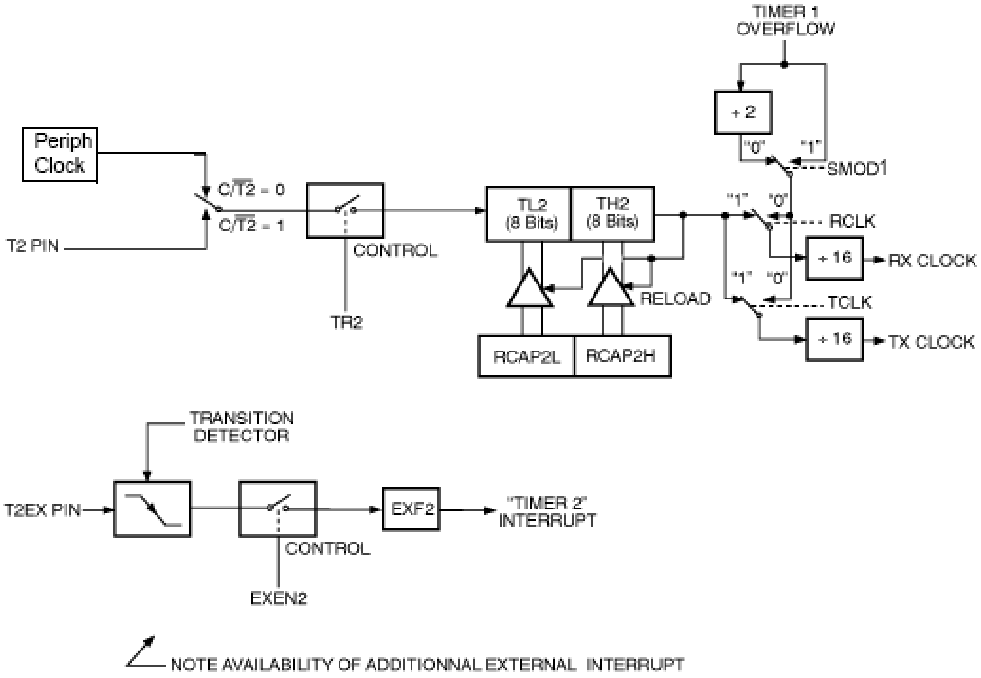
Table 2-16. Example of computed value when X2=0, SMOD1=0, SPD=0
Example of computed value when X2=0, SMOD1=0, SPD=0

Baud Rates	F_{OSCA} = 16.384 MHz		F_{OSCA} = 24 MHz	
	BRL	Error (%)	BRL	Error (%)
4800	247	1.23	243	0.16
2400	238	1.23	230	0.16
1200	220	1.23	204	0.16
600	185	0.16	152	0.16

Using Timer 2 to Generate Baud Rates

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2-9). Note then the baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 2-22.

Figure 2-22. Timer 2 in Baud Rate Generator Mode.



The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

Now, the baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate as follows:

The Timer can be configured for either "timer" or "counter" operation. In the most typical applications, it is configured for "timer" operation (C/$\overline{\text{T2}}$ = 0). "Timer" operation is a little different for Timer 2 when it's being used as a baud rate generator. Separately as a timer it would increment every machine cycle (thus at 1/12 the oscillator frequency). As a baud rate generator, however, it increment every state time (thus at 1/2 the oscillator frequency).

Timer 2 as a baud rate generator is shown in Figure 2-22. This Figure is valid only if RCLK + TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2, and will not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt, if desired.

It should be noted that when Timer 2 is running (TR2 = 1) in "Timer" function in the baud rate generator mode, one should not try to read or write TH2 or TL2. Under these conditions the Timer is being incremented every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but shouldn't be written to, because a write might overlap a reload and cause write and/or reload errors. In this case, turn the Timer off (clear TR2) before accessing the Timer 2 or RCAP registers.

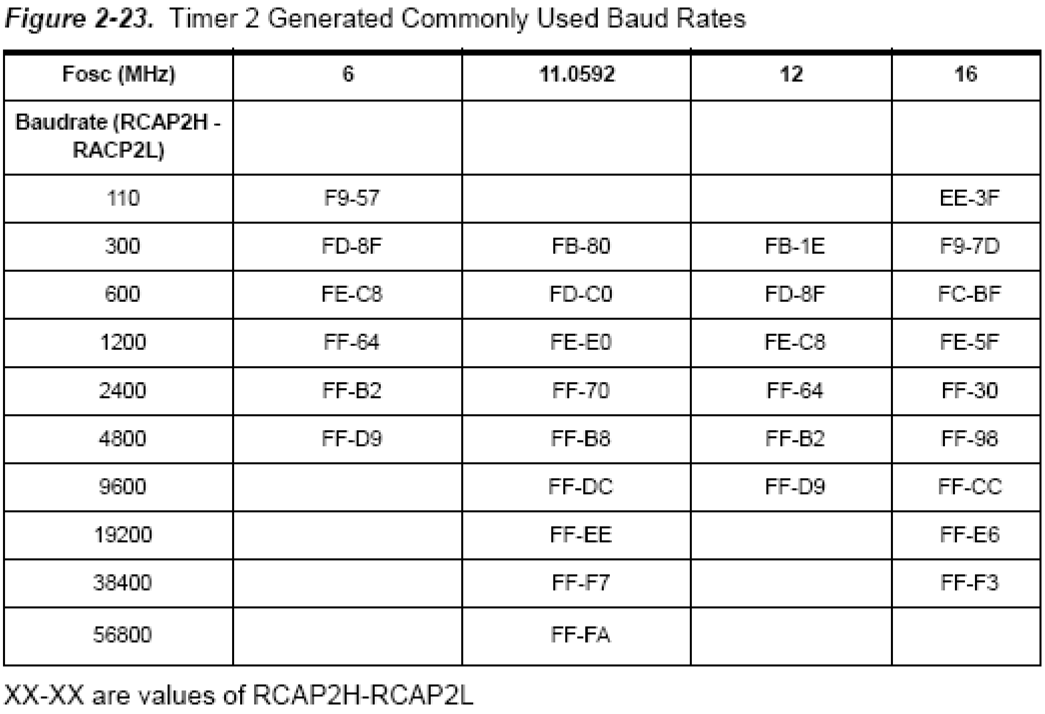


Figure 2-23. Timer 2 Generated Commonly Used Baud Rates

Fosc (MHz)	6	11.0592	12	16
Baudrate (RCAP2H - RACP2L)				
110	F9-57			EE-3F
300	FD-8F	FB-80	FB-1E	F9-7D
600	FE-C8	FD-C0	FD-8F	FC-BF
1200	FF-64	FE-E0	FE-C8	FE-5F
2400	FF-B2	FF-70	FF-64	FF-30
4800	FF-D9	FF-B8	FF-B2	FF-98
9600		FF-DC	FF-D9	FF-CC
19200		FF-EE		FF-E6
38400		FF-F7		FF-F3
56800		FF-FA		

XX-XX are values of RCAP2H-RCAP2L

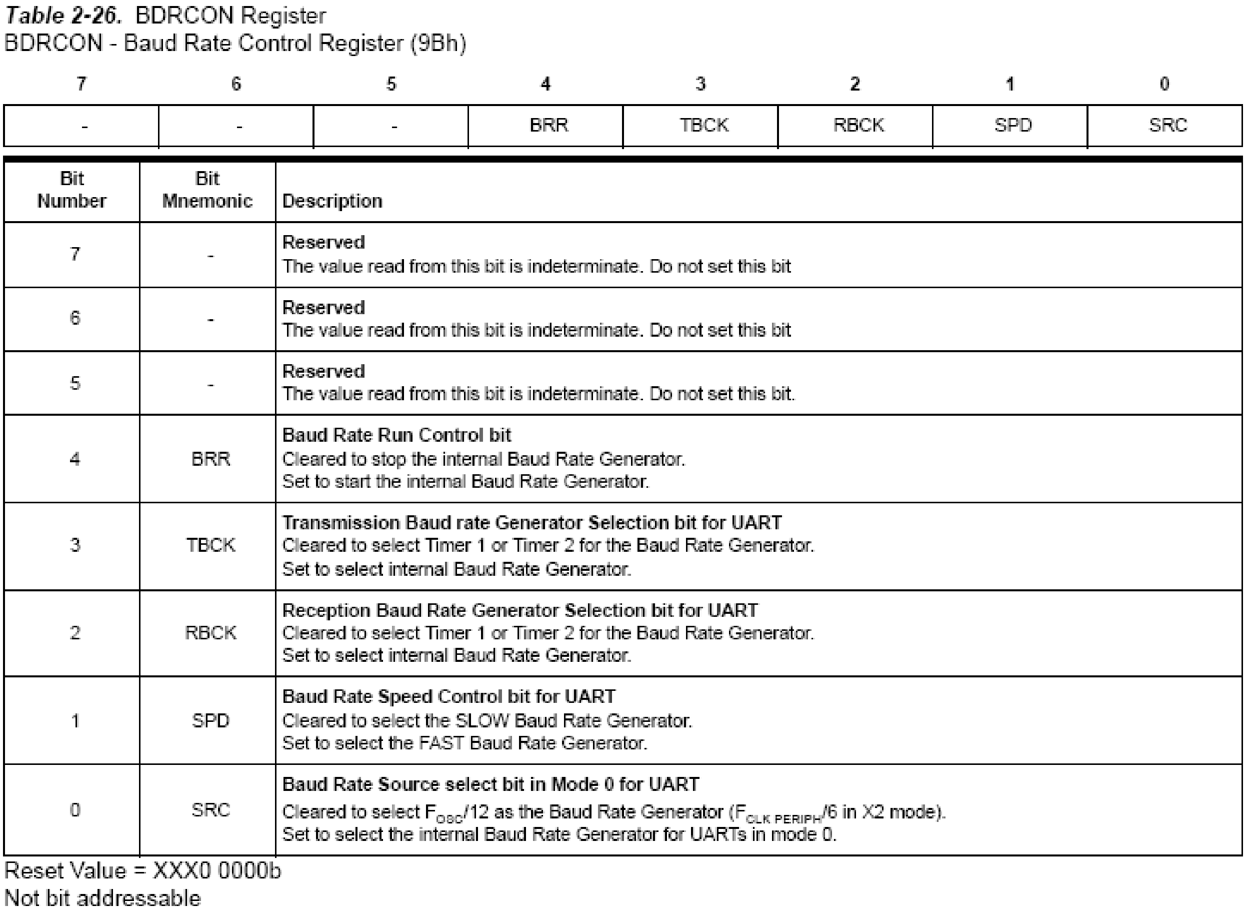


Table 2-26. BDRCON Register
BDRCON - Baud Rate Control Register (9Bh)

7	6	5	4	3	2	1	0
-	-	-	BRR	TBCK	RBCK	SPD	SRC

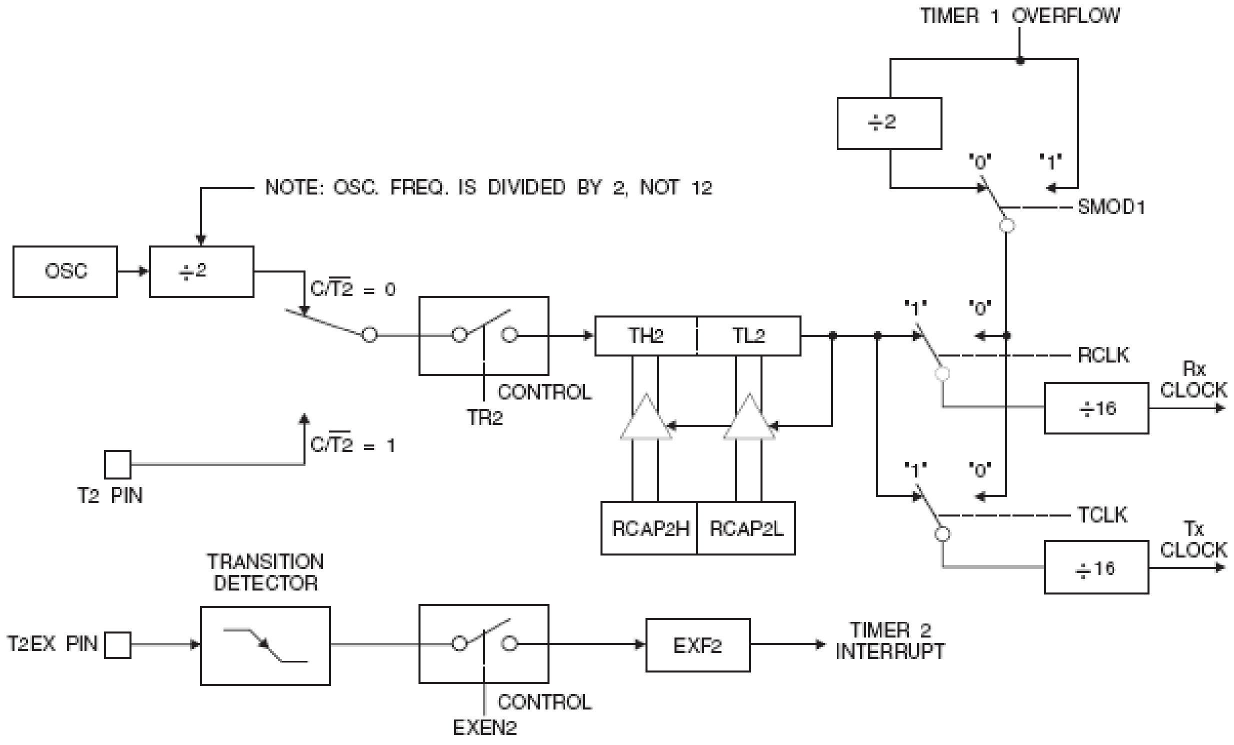
Bit Number	Bit Mnemonic	Description
7	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit
6	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit
5	-	**Reserved** The value read from this bit is indeterminate. Do not set this bit.
4	BRR	**Baud Rate Run Control bit** Cleared to stop the internal Baud Rate Generator. Set to start the internal Baud Rate Generator.
3	TBCK	**Transmission Baud rate Generator Selection bit for UART** Cleared to select Timer 1 or Timer 2 for the Baud Rate Generator. Set to select internal Baud Rate Generator.
2	RBCK	**Reception Baud Rate Generator Selection bit for UART** Cleared to select Timer 1 or Timer 2 for the Baud Rate Generator. Set to select internal Baud Rate Generator.
1	SPD	**Baud Rate Speed Control bit for UART** Cleared to select the SLOW Baud Rate Generator. Set to select the FAST Baud Rate Generator.
0	SRC	**Baud Rate Source select bit in Mode 0 for UART** Cleared to select F_{OSC}/12 as the Baud Rate Generator ($F_{CLK\ PERIPH}$/6 in X2 mode). Set to select the internal Baud Rate Generator for UARTs in mode 0.

Reset Value = XXX0 0000b
Not bit addressable

Table 10-1. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

·. Timer 2 in Baud Rate Generator Mode



Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 10-2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 10-4. The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software. The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the fol-lowing equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/T2 = 0). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

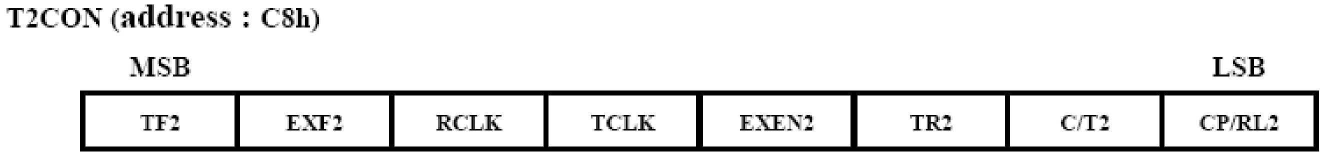
where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 10-4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt. Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

```
-------------------------
#define MSB_reload_value 0x36 /* msb reload value exemple */
#define LSB_reload_value 0x36 /* lsb reload value exemple */


$INCLUDE   (reg_c51.INC)

org 000h
ljmp begin

org 02Bh
ljmp it_timer2
;/**
; * FUNCTION_PURPOSE: This file set up timer 2 in mode 1 (clock-out mode and
; * negative transition detector).
; * The 16-bits register consist of all 8 bits of TH2 and all 8 bits of TL2.
; * TF2 does not generate interrupt.
; * A negative transition on P1.1(T2EX) generate an interrupt.
; * FUNCTION_INPUTS: void
; * FUNCTION_OUTPUTS: P1.0(T2) as clock output : Fout = Fperiph / (2*(65536-RCAP2)).
; */
org 0100h

begin:
  ANL T2MOD,#0FEh;           /* T2OE=1;DCEN=0; */
  ORL T2MOD,#02h;

  CLR EXF2;                            /* reset flag */
  CLR TCLK;
  CLR RCLK;                           /* disable baud rate generator */

  SETB EXEN2;                          /* enable events on T2EX */
  MOV TH2,MSB_reload_value;            /* Init msb_value */
  MOV TL2,LSB_reload_value;            /* Init lsb_value */
  MOV RCAP2H,MSB_reload_value     ;/* reload msb_value */
  MOV RCAP2L,LSB_reload_value     ;/* reload lsb_value */
  CLR C_T2;                            /* timer mode */
  CLR CP_RL2;                         /* reload mode */
  SETB EA;                             /* interupt enable */
  SETB ET2;                            /* enable timer2 interrupt */
  SETB TR2;                           /* timer2 run */

              JMP $;                  /* endless  */


;/**
; * FUNCTION_PURPOSE: timer2 interrupt
; * FUNCTION_INPUTS: void
; * FUNCTION_OUTPUTS: P1.2 toggle period = 2 *  P1.1(T2EX) period
; */
it_timer2:
              CLR TF2;                /* reset  interrupt flag */
              CPL P1.2;               /* P1.2 toggle when interrupt. */
  RETI

end
```

Using Timer 2 to Generate Baud Rates

Timer2 is selected as the baudrate generator by setting TCLK and/or RCLK in T2CON register as followed.



T2CON (address : C8h)

MSB LSB

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

T2CON.7: TF2 Timer2 overflow flag set by timer2 overflow and must be cleared by software. TF2 will not be set when
either RCLK=1 or TCLK=1.

T2CON.6: EXF2 Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2=1. when timer2 interrupt is enabled, EXF2=1 will cause the CPU to vector to the timer2 interrupt routine. EXF2 must be cleared by software.

T2CON.5: RCLK Receive clock flag. When set, cause the serial port to use timer2 overflow pulses for its receive clock in mode 1 and 3. RCLK=0 causes timer1 overflow to be used for the receive clock

T2CON.4: TCLK Transmit clock flag. When set, cause the serial port to use timer2 overflow pulses for its transmit clock in mode 1 and 3. TCLK=0 causes timer1 overflow to be used for the transmit clock

T2CON.3: EXEN2 Timer2 external enable flag. When set, allows a capture or reload to occur as a result of a negative
transition on T2EX if timer2 is not being used to clock the serial port. EXEN2=0 causes timer2 to ignore events at T2EX.

T2CON.2: Start/stop control for timer2. A logic 1 starts the timer

T2CON.1: Timer or counter select. (Timer 2) , 0 as internal timer

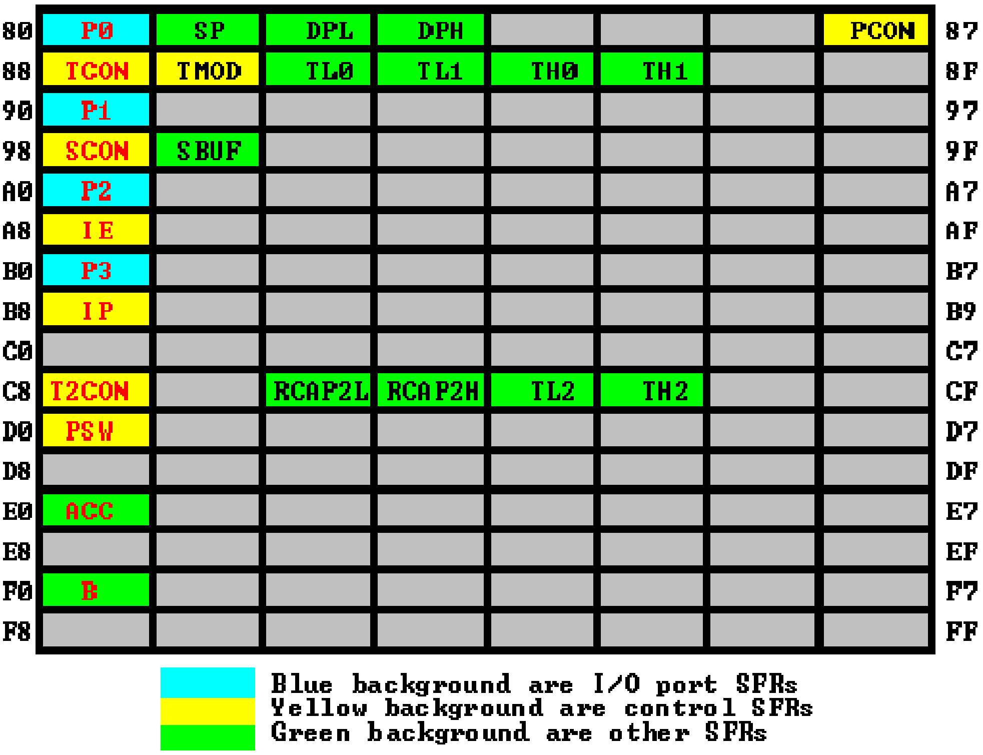
T2CON.0: Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2=1. When cleared, auto reloads will occur either with timer2 overflow or negative transitions at T2EX when EXEN2=1. When either RCLK=1 or TCLK=1, this bit is ignored and the timer is forced to auto-reload on timer2 overflow.

Note then the baudrates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer2 into its baudrate generator mode.

The baudrate generator mode is similar to the auto reload mode, in that a rollover is TH2 causes the Timer2 registers to be reload with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software given by the formula.
Baudrate= (Timer2 overflow rate)/16 =(Oscillator Frequency) / (32*(65536-(RCAP2H,RCAP2L)))

NEW SFRs FOR 8052'S THIRD TIMER

In addition to the 8051's standard 21 SFRs, the 8052 adds an additional 5 SFRs related to the 8052's third timer. All of the original 8051 SFRs function exactly as they do in the 8051-the 8052 simply adds new SFRs, it doesn't change the definition of the standard SFRs.
The five new SFRs are in the range of C8h to CDh (SFR C9h is not defined).



80	P0	SP	DPL	DPH				PCON	87
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
90	P1								97
98	SCON	SBUF							9F
A0	P2								A7
A8	IE								AF
B0	P3								B7
B8	IP								B9
C0									C7
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2			CF
D0	PSW								D7
D8									DF
E0	ACC								E7
E8									EF
F0	B								F7
F8									FF

Blue background are I/O port SFRs
Yellow background are control SFRs
Green background are other SFRs

T2CON SFR

The operation of Timer 2 (T2) is controlled almost entirely by the T2CON SFR, at address C8h. Note that since this SFR is evenly divisible by 8 that it is bit-addressable.

BIT	NAME	BIT ADDRESS	DESCRIPTION
7	TF2	CFh	**Timer 2 Overflow**. This bit is set when T2 overflows. When T2 interrupt is enabled, this bit will cause the interrupt to be triggered. This bit will not be set if either TCLK or RCLK bits are set.
6	EXF2	CEh	**Timer 2 External Flag**. Set by a reload or capture caused by a 1-0 transition on T2EX (P1.1), but only when EXEN2 is set. When T2 interrupt is enabled, this bit will cause the interrupt to be triggered.
5	RCLK	CDh	**Timer 2 Receive Clock**. When this bit is set, Timer 2 will be used to determine the serial port receive baud rate. When clear, Timer 1 will be used.
4	TCLK	CCh	**Timer 2 Receive Clock**. When this bit is set, Timer 2 will be used to determine the serial port transmit baud rate. When clear, Timer 1 will be used.
3	EXEN2	CBh	**Timer 2 External Enable**. When set, a 1-0 transition on T2EX (P1.1) will cause a capture or reload to occur.
2	TR2	CAh	**Timer 2 Run**. When set, timer 2 will be turned on. Otherwise, it is turned off.
1	C/T2	C9h	**Timer 2 Counter/Interval Timer**. If clear, Timer 2 is an interval counter. If set, Timer 2 is incremented by 1-0 transition on T2 (P1.0).
0	CP/RL2	C8h	**Timer 2 Capture/Reload**. If clear, auto reload occurs on timer 2 overflow, or T2EX 1-0 transition if EXEN2 is set. If set, a capture will occur on a 1-0 transition of T2EX if EXEN2 is set.

TIMER 2 AS A BAUD-RATE GENERATOR

Timer 2 may be used as a baud rate generator. This is accomplished by setting either RCLK (T2CON.5) or TCLK (T2CON.4).

With the standard 8051, Timer 1 is the only timer which may be used to determine the baud rate of the serial port. Additionally, the receive and transmit baud rate must be the same.

With the 8052, however, the user may configure the serial port to receive at one baud rate and transmit with another. For example, if RCLK is set and TCLK is cleared, serial data will be received at the baud rate determined by Timer 2 whereas the baud rate of transmitted data will be determined by Timer 1.

Determining the auto-reload values for a specific baud rate is discussed in Serial Port Operation; the only difference is that in the case of Timer 2, the auto-reload value is placed in RCAP2H and RCAP2L, and the value is a 16-bit value rather than an 8-bit value.

> **NOTE**: When Timer 2 is used as a baud rate generator (either TCLK or RCLK are set), the Timer 2 Overflow Flag (TF2) will not be set.

TIMER 2 IN AUTO-RELOAD MODE

The first mode in which Timer 2 may be used is Auto-Reload. The auto-reload mode functions just like Timer 0 and Timer 1 in auto-reload mode, except that the Timer 2 auto-relaod mode performs a full 16-bit reload (recall that Timer 0 and Timer 1 only have 8-bit reload values). When a reload occurs, the value of TH2 will be reloaded with the value contained in RCAP2H and the value of TL2 will be reloaded with the value contained in RCAP2L.

To operate Timer 2 in auto-reload mode, the CP/RL2 bit (T2CON.0) must be clear. In this mode, Timer 2 (TH2/TL2) will be reloaded with the reload value (RCAP2H/RCAP2L) whenever Timer 2 overflows; that is to say, whenever Timer 2 overflows from FFFFh back to 0000h. An overflow of Timer 2 will cause the TF2 bit to be set, which will cause an interrupt to be triggered, if Timer 2 interrupt is enabled. Note that TF2 will not be set on an overflow condition if either RCLK or TCLK (T2CON.5 or T2CON.4) are set.

Additionally, by also setting EXEN2 (T2CON.3), a reload will also occur whenever a 1-0 transition is detected on T2EX (P1.1). A reload which occurs as a result of such a transition will cause the EXF2 (T2CON.6) flag to be set, triggering a Timer 2 interrupt if said interrupt has been enabled.

TIMER 2 IN CAPTURE MODE

A new mode specific to Timer 2 is called "Capture Mode." As the name implies, this mode captures the value of Timer 2 (TH2 and TL2) into the capture SFRs (RCAP2H and RCAP2L). To put Timer 2 in capture mode, CP/RL2 (T2CON.0) must be set, as must be EXEN2 (T2CON.3).

When configured as mentioned above, a capture will occur whenever a 1-0 transition is detected on T2EX (P1.1). At the moment the transition is detected, the current values of TH2 and TL2 will be copied into RCAP2H and RCAP2L, respectively. At the same time, the EXF2 (T2CON.6) bit will be set, which will trigger an interrupt if Timer 2 interrupt is enabled.

> **NOTE 1**: Note that even in capture mode, an overflow of Timer 2 will result in TF2 being set and an interrupt being triggered.
>
> **NOTE 2**: Capture mode is an efficient way to measure the time between events. At the moment that an event occurs, the current value of Timer 2 will be copied into RCAP2H/L. However, Timer 2 will not stop and an interrupt will be triggered. Thus your interrupt routine may copy the value of RCAP2H/L to a temporary holding variable without having to stop Timer 2. When another capture occurs, your interrupt can take the difference of the two values to determine the time transpired. Again, the main advantage is that you don't have to stop timer 2 to read its value, as is the case with timer 0 and timer 1.

TIMER 2 INTERRUPT

As is the case with the other two timers, timer 2 can be configured to trigger and interrupt. In fact, the text above indicates a number of situations that can trigger a timer 2 interrupt.

To enable Timer 2 interrupt, set ET2 (IE.5). This bit of IE is only valid on an 8052. Similarly, the priority of Timer 2 interrupt can be configured using PT2 (IP.5). As always, be sure to also set EA (IE.7) when enabling any interrupt.

Once Timer 2 interrupt has been enabled, a Timer 2 interrupt will be triggered whenever TF2 (T2CON.7) or EXF2 (T2CON.6) are set. The Timer 2 Interrupt routine must be placed at 002Bh in code memory.

> **NOTE**: Like the Serial Interrupt, Timer 2 interrupt does not clear the interrupt flag that triggered the interrupt. Since there are two conditions that can trigger a Timer 2 interrupt, either TF2 or EXF2 being set, the microcontroller leaves the flags alone so that your interrupt routine can determine the source of the interrupt and act accordingly. It is possible (and even probable!) that you will want to do one thing when the timer overflows and something completely different when a capture or reload is triggered by an external event. Thus, be sure to always clear TF2 and EXF2 in your Timer 2 Interrupt. Failing to do so will cause the interrupt to be triggered repeatedly until the bits are cleared.

```
------------------
; Initialization
;
        ORG 0100h
;
START: MOV IE, #0              ; Disable ALL interrupts
;
;                              Set up timer 2
;
        MOV T2CON, #030h       ; Timer 2 : auto-reload mode for
                               ; both receive & transmit baud clocks
        MOV RCAP2L, #T2RL96L   ; Establish 16-bit reload value for 9600
        MOV RCAP2H, #T2RL96H
        MOV TL2, #T2RL96L      ; Make first timeout correct
        MOV TH2, #T2RL96H
```

----

2 datapointere

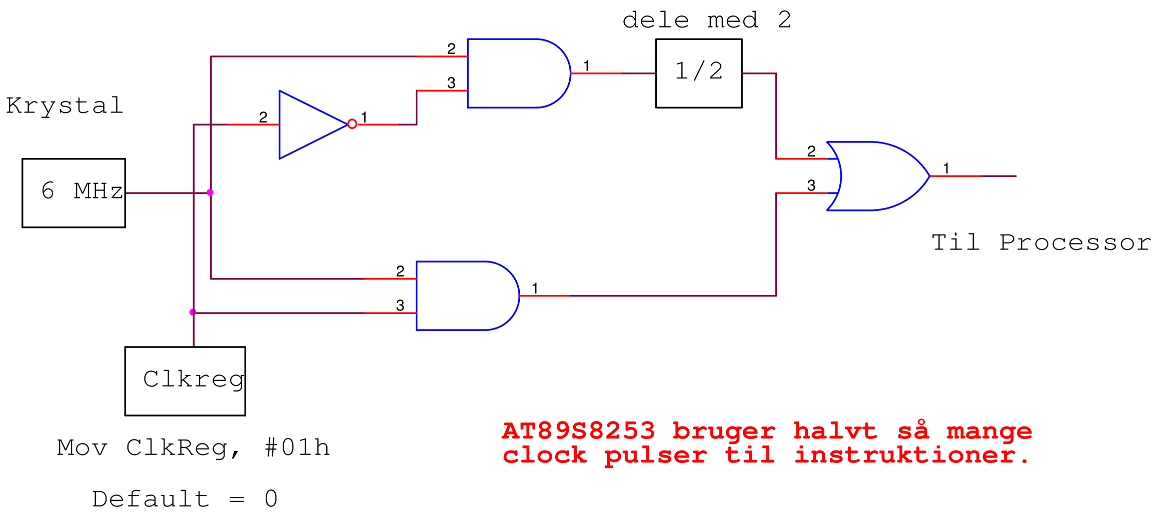
2 Data Pointere

Der er indbygget 2 stk 16 bit datapointere. DP0 på adresse 82h og 83h, og DP1 på 84h og 85h Vælges i SFR-register EECON. Bit DPS = 0 vælger DP0 (datapointer 0), DPS = 1 vælger DP1 Datapointer 1).

The AT89S8253 features two 16-bit data pointers (DP0 and DP1) for accessing data in program
memory, and on-chip EEPROM.

```
EECON DATA 96H             ; Watchdog and memory control register


DPS EQU 00000100B          ; data pointer select bit

xrl EECON, #DPS            ; switch data pointers
mov dptr, #XD_ADDR         ; pointer to external data memory
```

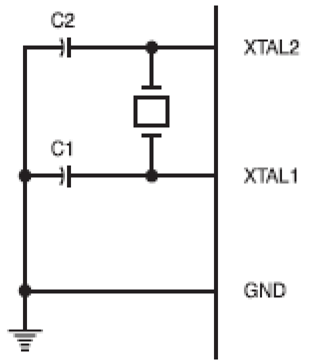
Krystal



AT89S8253 bruger halvt så mange clock pulser til instruktioner.

/Valle, Marts 07

Figure 16-1. Oscillator Connections



Note: C1, C2 = 5 pF ± 5 pF for Crystals

Den gamle 8252 brugte 22 pF, men nu er 5 pF anbefalet. Kan måske køre helt uden.

8253 er dobbelt så hurtig som den normale 8051 core. Hver instruktion tager 6 clock-cykler, men intern er clockfrekvensen delt med 2.

Clock register

Adr 8Fh

Mov CLKREG, #01h ; Herved divideres oscillatorfrekvensen ikke længere med 2. Så vil en 6 MHz krystal køre lige så hurtig som en 12 MHz, men det udsender mindre EMI.

Resetværdi er 0. Divideren er koblet ind.

Men i register CLKREG på adresse 08Fh, bit 0, X2, kan bruges til at udkoble denne divider.

Reset værdi af X2 er nul, og divideren er koblet ind.

```
CLKREG equ 08Fh
Mov a, #01h
Mov CLKREG, a            ; X2=1
```

Intern frekvens er max 24 MHz. Derfor bør krystallet max være 12 MHz.

Power On Reset

Der er indbygget power on reset. Derfor kan kondensatoren undværes !!
Reset input. Resetter på et højt. Der er indbygget power-on reset.

Watchdog Timer

Watchdog timeren tæller instruktions cyckler. Hvis watchdog timeren timer out, genererer den en intern reset til processoren. Bruges fx hvis man vil have processoren til at resette, hvis der ”køres i skoven”. I de subrutiner, der køres regelmæssigt under normale forhold, anbringes en ”Reset watchdog timer” instruktion ??????. Dvs, at under normale forhold sker der ingen time out. !

????? Hvordan arbejder den ??

Kilder:

Atmel Hardware bog http://www.atmel.com/dyn/resources/prod_documents/doc4316.pdf