SERIAL PORT PROGRAMMING

Data transfer between the PC and an 8051 system without any error is possible, if the baud rate of the 8051 system matches the baud rate of the PC's COM port.

BAUD RATE IN THE 8051

The 8051 transfers and receives data serially at many different baud rates. Serial communications of the 8051 is established with PC through the COM port. It must make sure that the baud rate of the 8051 system matches the baud rate of the PC's COM port/ any system to be interfaced. The baud rate in the 8051 is programmable. This is done with the help of Timer. When used for serial port, the frequency of timer is determined by (XTAL/12)/32 and 1 bit is transmitted for each timer period.

The Relationship between the crystal frequency and the baud rate in the 8051 is that the 8051 divides the crystal frequency by 12 to get the machine cycle frequency which is shown in Figure 5.2.1. Here the oscillator is XTAL = 11.0592 MHz, the machine cycle frequency is 921.6 kHz. 8051's UART divides the machine cycle frequency of 921.6 kHz by 32 once more before it is used by Timer 1 to set the baud rate. 921.6 kHz divided by 32 gives 28,800 Hz. Timer 1 in mode 2 is used to set the baud rate.

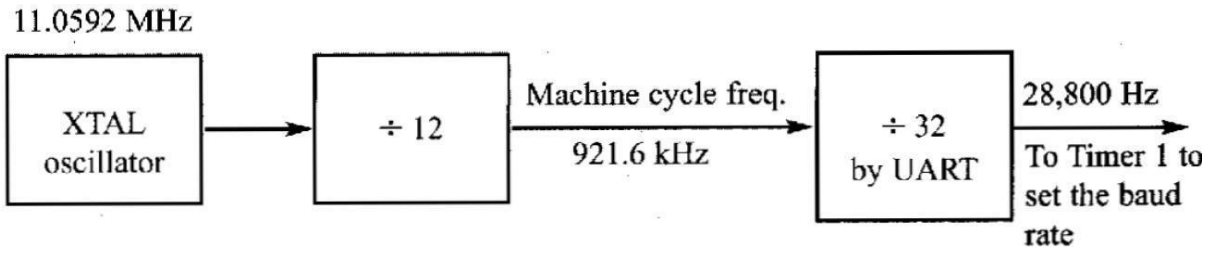

Figure 5.2.1 Frequency required to set the Baud rate

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, pg.no.288]

CALCULATION OF BAUD RATE:

In serial communication if data transferred with a baud rate of 9600 and XTAL used is 11.0592 MHz, then following steps to be followed to find the TH1 value to be loaded.

Clock frequency of timer clock: f = (11.0592 MHz / 12)/32 = 28,800Hz

Time period of each clock tick: T0 = 1/f = 1/28800

Duration of timer : n*T0 (n is the number of clock ticks)

9600 baud ->duration of 1 symbol: 1/9600

1/9600 = n*T0 = n*1/28800 n = f/9600

= 28800/9600 = 3 ->TH1 =-3

Similarly, for baud 2400 n = f/2400 =

12 ->TH1 = -12

BAUD RATE SELECTION

Baud rate is selected by timer1 and when Timer 1 is used to set the baud rate it must be programmed in mode 2 that is 8-bit, auto-reload. To get baud rates compatible with the PC, we must load TH1 with the values shown in Table 5.2.1.

Baud Rate	TH1 (Decimal)	TH1 (Hex)
9600	–3	FD
4800	–6	FA
2400	–12	F4
1200	–24	E8

Note: XTAL = 11.0592 MHz.

Table 5.2.1 Timer 1 THI register values for different baud rates

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, pg.no.287]

REGISTERS FOR SERIAL COMMUNICATION SBUF (SERIAL BUFFER) REGISTER:

It is an 8 bit register used solely for serial communication in the 8051.A byte of data to be transferred via the TxD line must be placed in the SBUF register. SBUF holds the byte of data when it is received by the RxD line. It can be accessed like any other register.

When a byte is written, it is framed with the start and stop bits and transferred serially via the TxD pin and when the bits are received serially via RxD, it is deframed

by eliminating the stop and start bits, making a byte out of the data received, and then placing it in the SBUF.

SCON (SERIAL CONTROL) REGISTER:

It is an 8 bit register used to program start bit, stop bit, and data bits of data framing, among other things.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier
SM2	SCON.5	Used for multiprocessor communication. (Make it 0.)
REN	SCON.4	Set/cleared by software to enable/disable reception.
TB8	SCON.3	Not widely used.
RB8	SCON.2	Not widely used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the beginning of the stop bit in mode 1. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software.

Note: Make SM2, TB8, and RB8 = 0.

Figure 5.2.2 SCON Register

[Source: "The 8051Microcontroller and Embedded Systems: Using Assembly and C" by Mohamed Ali Mazidi, Janice Gillispie Mazidi, Rolin McKinlay, pg.no.289]

STEPS TO SEND DATA SERIALLY:

1.Set baud rate by loading TMOD register with the value 20H, this indicating timer 1 in mode 2 (8-bit auto-reload) to set baud rate

2. The TH1 is loaded with proper values to set baud rate for serial data transfer

3. The SCON register is loaded with the value 50H, indicating serial mode 1, where an 8- bit data is framed with start and stop bits

4. TR1 is set to 1 to start timer 1

5. TI is cleared by CLR TI instruction

6. The character byte to be transferred serially is written into SBUF register

7. The TI flag bit is monitored with the use of instruction JNB TI,xx to see if the character has been transferred completely

8. To transfer the next byte, go to step 5

Program to transfer letter "A" serially at 9800baud, continuously:

```
MOV TMOD,#20H          ; timer 1,mode 2(auto reload)
MOV TH1, #-3           ; 9600 baud rate
MOV SCON, #50H         ; 8-bit, 1 stop, REN enabled
SETB TR1               ; start timer 1
AGAIN: MOV SBUF, #"A"  ; letter "A" to transfer
HERE: JNB TI, HERE     ; wait for the last bit
CLR TI                 ;clear TI for next char
SJMP AGAIN             ; keep sending A
```

IMPORTANCE OF THE TI FLAG:

Check the TI flag bit, we know whether or not 8051 is ready to transfer another byte. TI flag bit is raised by the 8051 after transfer of data. TI flag is cleared by the programmer by instruction like "CLR TI". When writing a byte into SBUF, before the TI flag bit is raised, it may lead to loss of a portion of the byte being transferred

STEPS TO RECEIVE DATA SERIALLY:

1. Set baud rate by loading TMOD register with the value 20H, this indicating timer 1 in mode 2 (8-bit auto-reload) to set baud rate .
2. The TH1 is loaded with proper values to set baud rate
3. The SCON register is loaded with the value 50H, indicating serial mode 1, where an

8- bit data is framed with start and stop bits

4. TR1 is set to 1 to start timer 1
5. RI is cleared by CLR RI instruction
6. The RI flag bit is monitored with the use of instruction JNB RI,xx to see if an entire character has been received yet
7. When RI is raised, SBUF has the byte; its contents are moved into a safe place 8. To receive next character, go to step 5

Program to receive bytes of data serially, and put them in P2, set the baud rate at 9600, 8-bit data, and 1 stop bit:

```
MOV TMOD, #20H      ; timer 1,mode 2(auto reload)
MOV TH1, #-3        ; 9600 baud rate
MOV SCON, #50H      ; 8-bit, 1 stop, REN enabled
SETB TR1            ; start timer 1
HERE: JNB RI, HERE  ; wait for char to come in
MOV A, SBUF         ; saving incoming byte in A
MOV P2, A           ; send to port 1
CLR RI              ; get ready to receive next byte
SJMP HERE           ; keep getting data
```

IMPORTANCE OF THE RI FLAG BIT:

It receives the start bit, next bit is the first bit of the character about to be received. When the last bit is received, a byte is formed and placed in SBUF. when stop bit is received, it makes RI = 1 indicating entire character byte has been received and can be read before overwritten by next data. When RI=1, received byte is in the SBUF register, copy SBUF contents to a safe place. After the SBUF contents are copied the RI flag bit must be cleared to 0.