SERIAL PORT OF 8051

8051 has a **high speed**, **full duplex**, software **programmable** Serial Port.
Data is **received** serially through the **R_xD** line, and **transmitted** through the **T_xD** line.
SBUF register acts as a **buffer for both** reception and transmission.
The **SCON** SFR mainly **controls** serial **Communication**.
The **SMOD bit in** the **PCON** SFR **controls** the **baud rate**.

SCON - Serial Control (SFR) [Bit-Addressable As SCON.7 to SCON.0]

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0 and SM1: (Serial Port Mode Bits 0 and 1)

SM0	SM1	SERIAL MODE	DESCRIPTION	BAUD RATE
0	0	Mode 0	Shift Register	Fixed ➔ $f_{osc}/12$
0	1	Mode 1	8-bit UART	Variable
1	0	Mode 2	9-bit UART	Fixed ➔ $f_{osc}/32$ or $f_{osc}/64$
1	1	Mode 3	9-bit UART	Variable

SM2: (Serial Port Mode Bits 2)
Enables multiprocessor features in Mode 2 and Mode 3. i.e.
Mode 2 or Mode 3: If **SM2 = 1** **RI will be 1 when** the **9th data bit** received is **"1".**
Mode 1: If **SM2 = 1** **RI will be 1 when** the **Stop bit** received is **"1"**. (valid)
Mode 0: **SM2** is **kept 0** i.e. Not Used.

REN: (Receiver Enable)
REN = 1 Enables the Receiver.
REN = 0 Disables the Receiver.

TB8: (Bit 8 i.e. 9th bit transmitted)
In Mode 2 and Mode 3 it holds the 9th Programable bit, to be transmitted.
In Mode 1 and Mode 0 it is not used.

RB8: (Bit 8 i.e. 9th bit recieved)
In Mode 2 and Mode 3 it is used to receive the 9th Programable bit.
In Mode 1 it recieves the Stop bit.
In Mode 0 it is not used.

RI: (Receive Interrupt)
RI = 1 ➔ One Complete character is received.
RI must be explicitly cleared by software before receiving the next byte.

TI: (Transmit Interrupt)
TI = 1 ➔ One Complete character is transmitted.
TI must be explicitly cleared by software before transmitting the next byte.

The Baud Rate is controlled by the bit SMOD in PCON SFR.
SMOD: (Serial Baud rate Modify Bit in PCON SFR)
SMOD = 1 ➔ Doubles the Baud rate of Timer 1 for Modes 1, 2 and 3.
SMOD = 0 ➔ Uses Timer 1 Baud Rate.

SBUF Register

It is physically two registers, one for holding the received character and the other for holding the character to be transmitted.
Both receive and transmit registers are addressed 99H.

Serial data Interrupt

When a **complete character** is **received** the **RI** bit is **set** in the SCON Register.
When a **complete character** is **transmitted** the **TI** bit is **set** in the SCON Register.
These two bits are **OR'ed** to produce the serial data interrupt.
RI/TI bit must be **explicitly cleared** by the program before transferring the next charecter.

Data Transmission

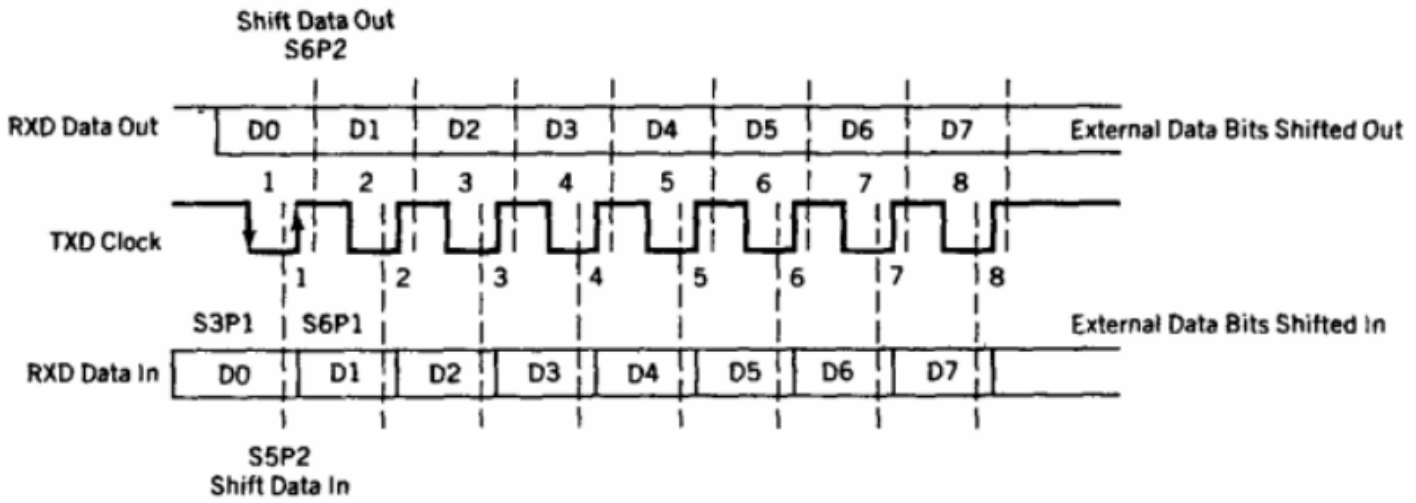
Data **written** in **SBUF**.
Data **transmitted** through **T_xD**.
When a complete character is transmitted **TI** is **set**.
Serial Data interrupt occurs.
In its **ISR** the **TI** bit is **reset**.
New data written **into SBUF** and the process **continues** till all the data is transmitted.

Data Reception

REN in SCON **must be set**, this prevents receiving unwanted data (eg noise).
Data **received** through **R_xD into** the **SBUF**.
When a complete character is received **RI** is **set**.
Serial Data interrupt occurs. #Please refer Bharat Sir's Lecture Notes for this ...
In its **ISR** the **RI** bit is **reset** and so the **program accepts the received data**.
New data is received **in SBUF** and the process **continues** till all the data is received.

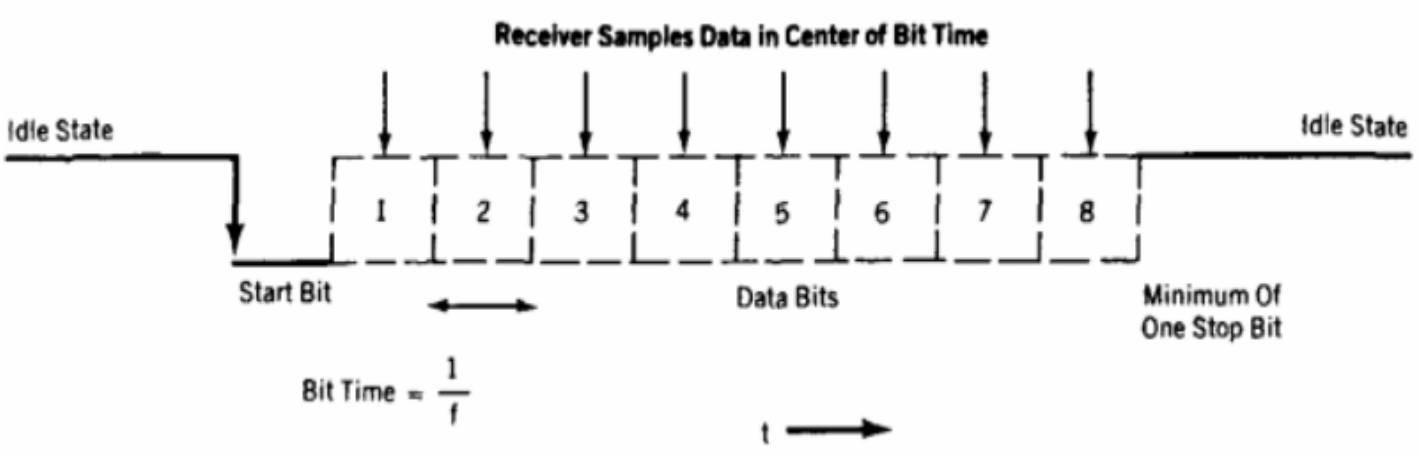
Serial Data Transfer Modes

a) Mode 0 (Shift Register)



This is a **8-bit Half-Duplex** mode.
The **Start/Stop** bits are **not required**.
Both **transmission and reception** happen through the **R_xD** line.
T_xD line **provides** the **shift clock** for data transfer.
The signal on the T_xD line is a square wave high for S_6, S_1, and S_2 and low for S_3, S_4, and S_5.
During **Tramsnission** data is **"shifted"** out of **SBUF** during S_6 P_2. #Please refer Bharat Sir's Lecture Notes for this ...
During **Reception R_xD** is **sampled** during S_5 P_2 **and then "shifted".**
This mode is **mainly used** for high **speed data collection** using discrete logic.
It is **not intended** for data communication **between computers**.
Baud rate: Fixed.
f_{Baud} = f/12. (f = Oscillator Frequency).

b) Mode 1 (8-bit "Standard" UART)



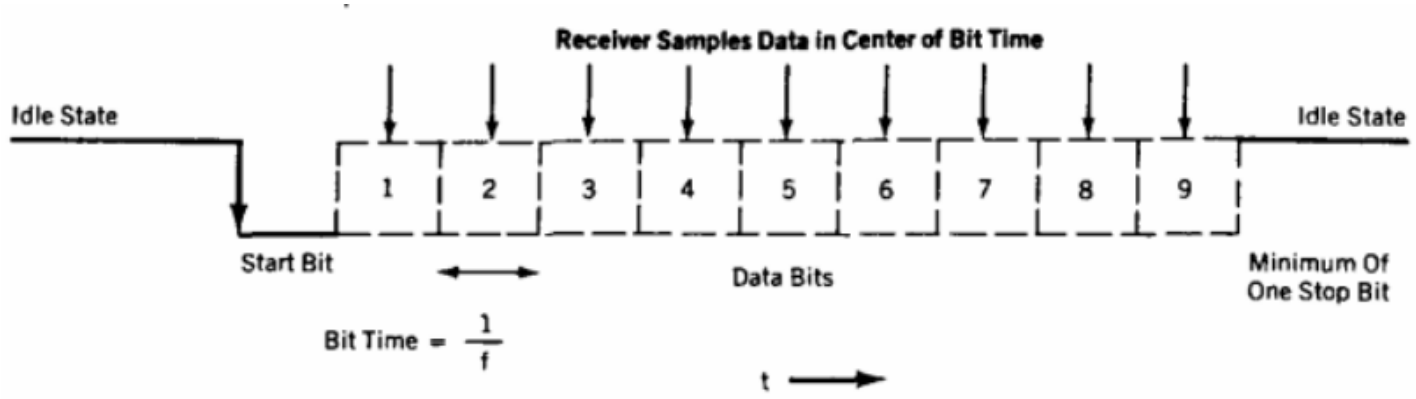
This is a **10-bit Full-Duplex** mode.
R_xD receives data, **T_xD transmits** data.
For Transmission data is sent as **1 Start** bit (0), **8 Data** bits (LSB first), **1 Stop** bit (1).
Transmit Interrupt flag **TI** is **set** only **after** all the above **10** bits are **transmitted**.
Each bit interval is inverse of Baud rate i.e. each bit has to be maintained for that interval.
Data is also **received** in the **same order** at the programmed Baud Rate.
During reception, **Start bit** is **discarded**, **8 Data** bits received in **SBUF**, **Stop** bit saved in **RB8**.
The **RI** will be set **only if SM2 = 0 (unconditional) or** the **RB8 = 1 (condition satisfied)**. This is an *anti-noise safeguard*.
Once RI is set the program is interrupted to accept the data just received.
Baud Rate: Variable (can be controlled be changing the overflow rate using difeent counts)

$$\mathbf{f_{Baud} = \frac{2^{SMOD}}{32} \times (Timer\ 1\ overflow\ frequency).}$$

$$f_{Baud} = \frac{1}{32} \times (\text{Timer 1 overflow frequency}); \{\text{when SMOD = 0}\}.$$

$$f_{Baud} = \frac{1}{16} \times (\text{Timer 1 overflow frequency}); \{\text{ when SMOD = 1}\}.$$

c) Mode 2 (9-bit UART) (Multiporcessor Mode)



This mode is similar to Mode 1 except that it has **11 bits** per character.
Format : **1 Start** bit (0), **8 Data** bits, **9th Programmable Bit** and **1 Stop** bit (1).
During **Transmission** the **9th** data bit is **copied** form bit **TB8** of SCON.
During **Reception** the **8 data bits** are **received in SBUF**, **9th** bit **in RB8** of SCON.
Both **Start** bit and **Stop** bits are **discarded**.
This **9th bit can** be used to **control Multiprocessor Communication**.
Baud Rate: Fixed (cannot be controlled be Timer1)

$$\mathbf{f_{Baud} = \frac{2^{SMOD}}{64} \times (Oscillator\ Frequency).}$$

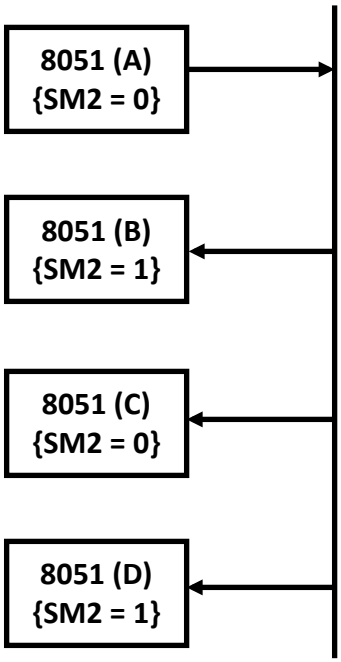
$$f_{Baud} = \frac{1}{64} \times (\text{Oscillator Frequency}) \{\text{when SMOD} = 0\}.$$

$$f_{Baud} = \frac{1}{32} \times (\text{Oscillator Frequency}) \{\text{when SMOD} = 1\}.$$

d) Mode 3 (9-bit UART) (Multiporcessor Mode)

This mode is exactly the **same as Mode 2 except** that the **Baud rate is Variable** i.e. it is determined using the Timer 1 overflow rate **as in Mode 1**.

Multiprocessor Communication in Mode 2**:**



This mode is also used for multiprocessor communication.
Here the 9th bit is very useful.

If the **9th** bit is **1** … Then **every processor** will be **interrupted** and hence will **recive** the data (SM2=0 or 1)
This can be use to **"Broadcast"** the data to all processors.

It the **9th** bit is **0** … Then **ONLY** those processors with **SM2 = 0** will be **interrupted**, hence **receive** the data.
This enables **Selective Transmission** i.e. processors can selectively talk to one another.

In the above example,
If "A" send data with the 9th bit as 1 then all will receive, and it will be a "Broadcast"
If "A" send data with the 9th bit as 0 then only "C" will receive, and hence it will be "Selective Transmission".