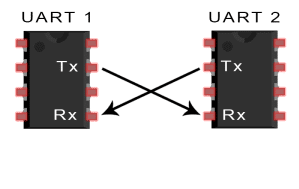
**UART**

**Introduction to UART:**

A UART(**Universal Asynchronous Receiver/Transmitter**) is a physical circuit in a microcontroller, or a stand-alone IC. A UART’s main purpose is to transmit and receive serial data.

In UART communication, two UARTs communicate directly with each other. The transmitting UART converts parallel data from a controlling device like a CPU into serial form, transmits it in serial to the receiving UART, which then converts the serial data back into parallel data for the receiving device. Only two wires are needed to transmit data between two UARTs. Data flows from the Tx pin of the transmitting UART to the Rx pin of the receiving UART.



UARTs transmit data asynchronously, which means there is no clock signal to synchronize the output of bits from the transmitting UART to the sampling of bits by the receiving UART. Instead of a clock signal, the transmitting UART adds start and stop bits to the data packet being transferred. These bits define the beginning and end of the data packet so the receiving UART knows when to start reading the bits. When the receiving UART detects a start bit, it starts to read the incoming bits at a specific frequency known as the baud rate. Baud rate is a measure of the speed of data transfer, expressed in bits per second (bps). Both UARTs must operate at about the same baud rate. Both UARTs must also must be configured to transmit and receive the same data packet structure.

**How UART Works?**

The UART that is going to transmit data receives the data from a data bus. The data bus is used to send data to the UART by another device like a CPU, memory, or microcontroller. Data is transferred from the data bus to the transmitting UART in parallel form. After the transmitting UART gets the parallel data from the data bus, it adds a start bit, a parity bit, and a stop bit, creating the data packet. Next, the data packet is output serially, bit by bit at the Tx pin. The receiving UART reads the data packet bit by bit at its Rx pin. The receiving UART then converts the data back into parallel form and removes the start bit, parity bit, and stop bits. Finally, the receiving UART transfers the data packet in parallel to the data bus on the receiving end.

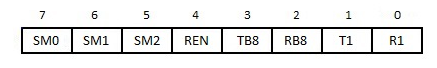
UART transmitted data is organized into packets. Each packet contains 1 start bit, 5 to 9 data bits (depending on the UART), an optional parity bit, and 1 or 2 stop bits.

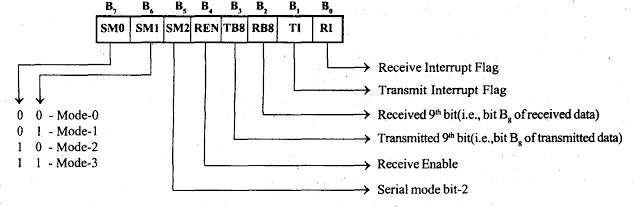
**8051 UART:**

The registers need to be configured for using 8051 UART peripheral are :

1. SCON(serial port control register)
2. SBUF (serial buffer)
3. PCON (power control register)
4. TMOD (Timer/Counter mode control register)
5. TCON (Timer/Counter control register)

**SCON**

****



**Mode 0:**

* In this mode the serial port function as half duplex serial port with fixed baud rate.
* The 8- bit serial data is received and transmitted through RxD pin and the controller output the shift clock through TxD pin during reception and transmission.
* The baud rate is fixed at 1 / 12 of the oscillator frequency.

**Mode 1:**

* In this mode the serial port function as full duplex serial port with variable baud rate.
* In this mode one data consists of 10 bits, which includes one start bit, eight data bit and one stop bit. During reception the stop bit is stored as RB8 in SCON register.
* Baud rate in mode-1 depends on the value of SMOD bit in PCON register and the

timer- 1overflow rate.

**Mode 2:**

* In this mode the serial port function as full duplex serial port with a baud rate of either 1/32 or 1/64 of the oscillator frequency.
* In this mode one data consists of 11 bits which includes one start bit, eight data bit, a programmable 9th data bit and one stop bit.
* During transmission the TB8 of SCON register is added as 9th data bit and during reception the 9th data bit is stored as RB8 in SCON register.
* The baud rate depends on the value of SMOD bit in PCON register.

**Mode 3:**

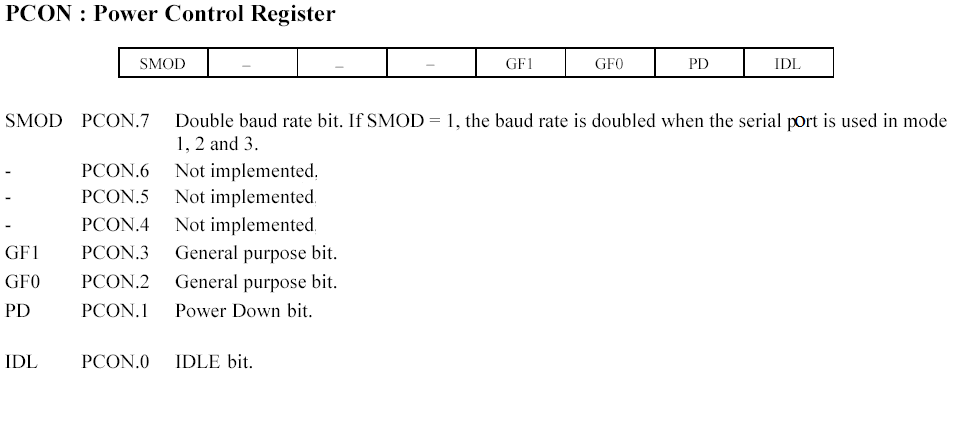
* The mode-3 is same as mode-2, except the baud rate.
* In mode-3, the baud rate is variable. The baud rate depends on the value of SMOD bit in PCON register and the timer- 1 overflow rate.
* The serial mode bit-2 (SM2) has no effect in serial mode-0 and mode 1.
* In mode-2 and mode-3 the SM2 bit is used to enable multiprocessor communication.
* In multiprocessor communication the serial port of a number of microcontrollers can be connected to a common serial bus. One controller will act as a master and all other controller will act as slave.

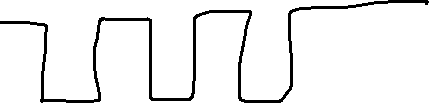
**REN:**

The REN bit of SCON register can be used to enable or disable the serial reception. When REN = 1, the serial reception is enabled and when REN = 0, the serial reception is disabled.

**TI & RI:**

* The bits TI and RI of SCON register are transmitting interrupt flag and receive interrupt flag respectively.
* The controller will set the TI bit during the transmission of stop bit of a data character in mode 1 to 3 and during the transmission last bit (most significant bit) of a data character in mode-0.
* The controller will set the RI bit during the reception of stop bit of a data character in mode 1 to 3 and during the reception of last bit of a data character in mode- 0.





BR = Number of symbols per second

Here in 8051, 1symbol is nothing but 1bit

Eg: 9600BR for PC, GSM

In 8051 =9600 =9600 bits per second

In Advanced controllers

1Symbol= 2/4/6/8 etc…

‘A’

8051 ---🡪 IBM PC

We know IBM PC is communicating through UART

We know, IBM PC BR=9600

In 8051 1 symbol = 1clock cycle/1bit

S0, 9600 bits per second

To transmit one Ascii value

1start bit 8 databits stopbit

0 ‘A’ 1 =10BITS for one ascii/byte

9600/10=960 bytes per second

960bytes ------🡪 1

1byte ------🡪 ?

960\*Y=1x1

Y=1/960 is the byte time period

Bit Time Period =Y/8

Why /8, because 1byte=8bits