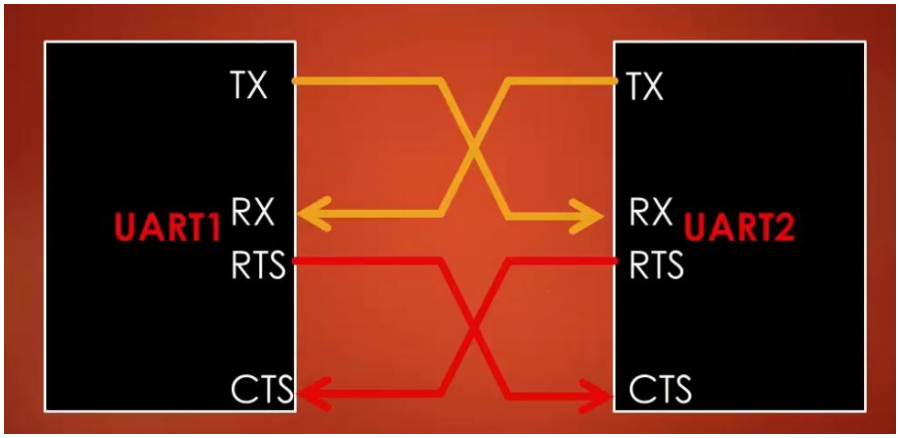
UART

- Stands for universal asynchronous receiver-transmitter
- data format and transmission speeds are configurable
- Communication may be *simplex , half duplex or full duplex*

UART PINS:

- RX
    - Uart baud continuously samples rx line to detect the start bit of the frame
    - Once start bit is received , it keeps on sampling frame until stop byte is received
- TX
    - If we are not sending anything , the tx line is held high which is idle state of tx line
- CTS
    - If hardware flow control is used
    - Stands for clear to send
    - Active low
    - If hardware flow control is used , data transmission happens only when CTS is low
    - If it is high , data transmission is held till it is high
    - It has to be controlled by another device (externally triggered)
    - Connected to RTS of another device (uart)

- RTS
    - Stands for request to send
    - Active low pin
    - Device uses this pin to inform the other device that it needs the data
    - Connected to CTS of another device (uart)

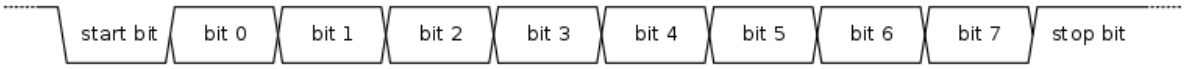


Data framing

A frame refers to entire data packet which is being received or sent during the communication.
Depending on the communication protocol format of fram may vary
Typical frame format has this order

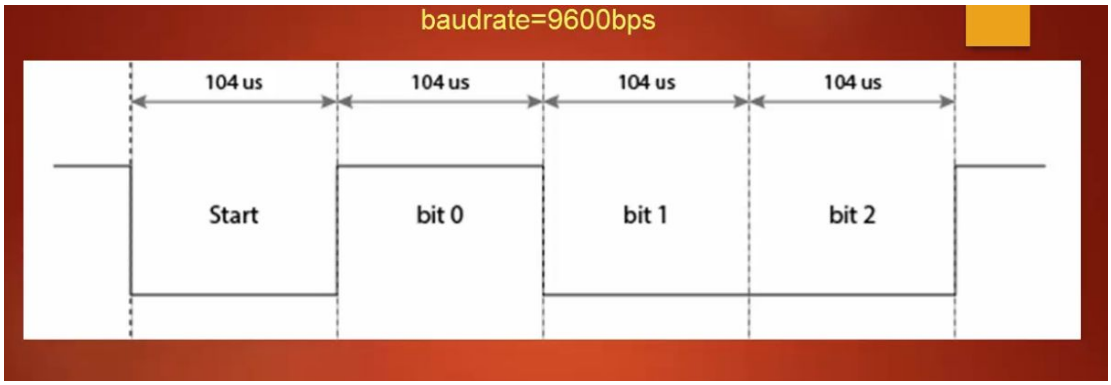
1. Start bit which a low for one bit duration (transition from high to low)
2. Then follows data bit from LSB to MSB which usually occupies five to nine bits.(configuration registers can be used to decide number of data bits per frame).
3. Then follows the parity bit (this is optional) , this bit denotes odd parity or even parity
4. Then comes the stop bit, it is logic high for a duration of one / one and half / two bits.

- takes bytes of data and transmits the individual bits in a sequential fashion
- contains a parallel in shift register
- Supports five to nine bits per data packet , depending on the code set employed
- If a parity bit is used, it would be placed after all of the data bits.
- The next one or two bits are stop stop bits



BAUD RATE:

- This symbolises how fast a data is being sent over a serial line
- Expressed in bits per second
- Inverting the baud rate gives how long it takes to transfer a single bit . ie how long the transmitter line holds the pin high or low
- Both transmitting and receiving device shall operate at same baud rate .
- Baud rate is dependent on clock frequency of microcontroller



Synchronisation bits:

Two or three bits that are being sent with each data packet

These are

- START BIT
  - Marks beginning of packet
  - There is only one start bit

    - Indicated by idle data line , going high to low

- STOP BIT
    - Marks end of packet
    - Configurable between one bit or one and half bit or two bits
    - indicated by idle data line , going low to high
    - For very high baud rate advisable to have two stop bits
    - Typically

PARITY BIT

Adding parity bit is a simplest method of error detection . parity is the number of ones in the binary number.

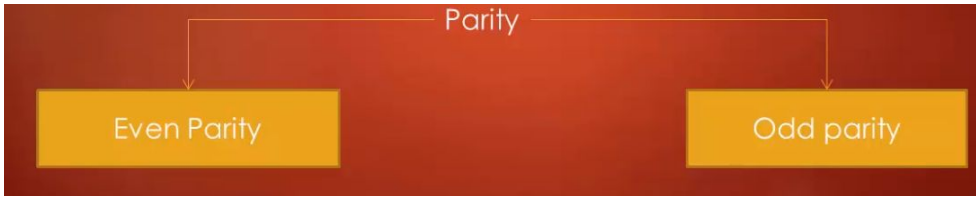
Even parity:

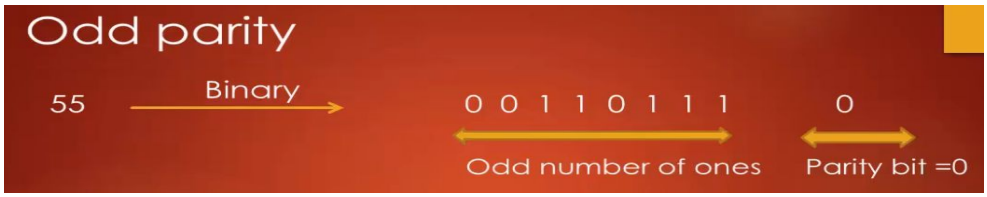
- Results in even number of ones when counted including the parity bit
- If the number of al highl bits in data excluding parity bit is odd , parity is set 1 , so that number of ones in data packet including parity bit becomes even
- If the number of al highl bits in data excluding parity bit is even , parity is set 0 , so that number of ones in data packet including parity bit becomes even

Odd parity

- Results in odd number of ones when counted including the parity bit
- If the number of al highl bits in data excluding parity bit is even , parity is set 1 , so that number of ones in data packet including parity bit becomes odd

- If the number of al highl bits in data excluding parity bit is odd , parity is set 0 , so that number of ones in data packet including parity bit becomes odd





Transmitter

Heart of transmitter is a parallel in serial out shift register , where parallel data is converted to serial data sequences . shift register obtains its data from transmit data register (this TDR register is loaded with the data by the software). The data is not put in shift register until stop byte form last data lot (payload) is received .

- As soon as the sending system deposits a character in the shift register ,the UART generates a start bit, shifts the required number of data bits out to the line
- generates and sends the parity bit (if used)
- sends the stop bits
- UART maintains a flag showing busy status so that the host system knows if there is at least one character in the transmit buffer or shift register (may also be signaled with an interrupt)

- Steps:
    - Program the register to define the word length (no of data bits per packet)
    - Program register to define number of stop bits
    - Program register/ registers to select the desired baud rate
    - Set bit to enable transmission block
    - Set bit to enable UART
    - If txe flag is set , write the data to be sent in uart tx register ,
    - Wait until transmission complete flag is set (which means last data transfer is successful) to one (all data written in tx data register)

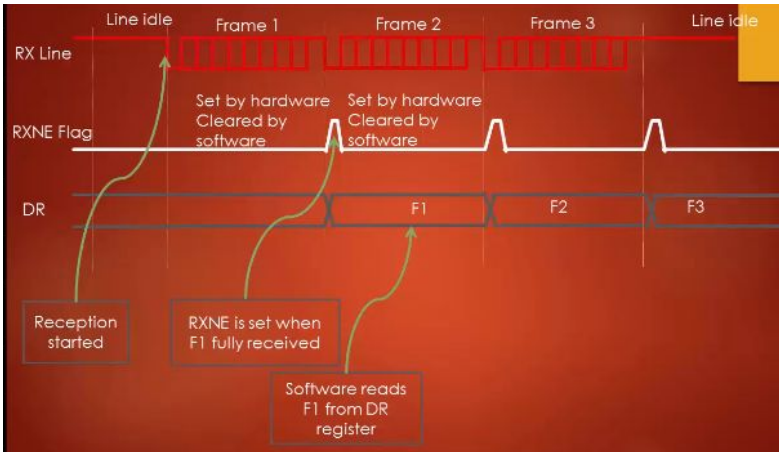
Receiver

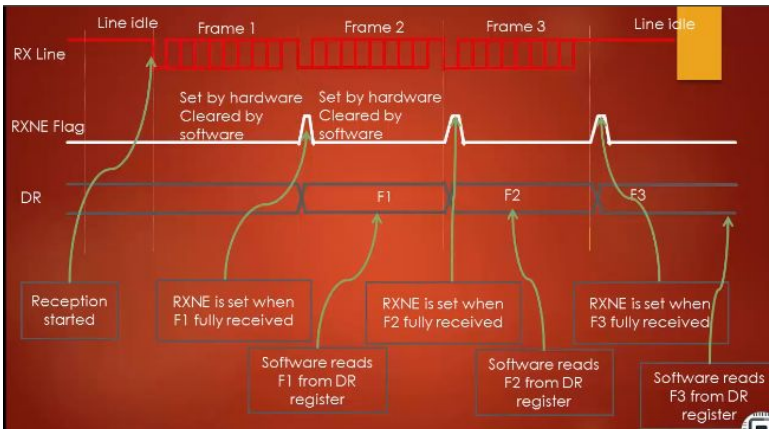
Heart of receiver is a serial in parallel out shift register which shifts LSB first, where serial sequence data is converted to parallel data . once stop bit is received shift register obtains its data from receive data register (this RDR register is loaded with the data by the hardware from receive shift register).

- All operations of the UART hardware are controlled by a clock signal which runs at a multiple of the data rate , typically 8 times the bit rate
- receiver tests the state of the incoming signal on each clock pulse , looking for the beginning of the start bit
- If the apparent start bit lasts at least one-half of the bit time, it is valid and signals
- After waiting a further bit time, the state of the line is again sampled and the resulting level clocked into a shift register

- After the required number of bit periods for the character length (5 to 8 bits, typically) have elapsed, the contents of the shift register are made available
- The UART will set a flag indicating new data is available, and may also generate a processor interrupt to request that the host processor transfers the received data.
- UARTs resynchronize their internal clocks on each change of the data line that is not considered a spurious pulse
- Simplistic UARTs resynchronize on the falling edge of the start bit only, and then read the center of each expected data bit
- Steps:
    - Program the register to define the word length (no of data bits per packet)
    - Program register to define number of stop bits
    - Program register/ registers to select the desired baud rate
    - Set bit to enable UART
    - Set bit to enable reception block (starts searching for start bit)

- Wait until reception complete flag is set (which means last data received is copied to uart receive buffer) to one (all data written in tx data register)
- If interrupt for receive complet flag is set , read the data received from uart rx register ,





Break condition

- occurs when the receiver input is at the "space" (logic low, i.e., '0') level for longer than some duration of time , typically, for more than a character time.
- appears to the receiver as a character of all zero bits with a framing error
- When signaling rates are mismatched, no meaningful characters can be sent, but a long "break" signal can be a useful way to get the attention of a mismatched receiver to do something (such as resetting itself).
- systems can use the long "break" level as a request to change the signaling rate

ERRORS :

- Overrun
    - occurs when the receiver cannot process the character that just came in before the next one arrives
- Underrun error
    - occurs when the UART transmitter has completed sending a character and the transmit buffer is empty.
    - indication that no data remains to be transmitted
    -
- Framing error
    - detect a *framing error* when it does not see a "stop" bit at the expected time after a "stop" bit
    - A "break" condition on the line is also signaled as a framing error.

- Parity error
    - occurs when the parity of the number of 1 bits disagrees with that specified by the parity bit
    - Use of a parity bit is optional, so this error will only occur if parity-checking has been enabled.