**Asynchronous Transmission**

* 
1. Asynchronous transmission is also known as ***start-stop mode*** or ***character mode*.**
2. Each character is *sent* as an independent unit of data that may be transmitted and received independently. However Data may also be transmitted as a continuous stream of characters.
3. Asynchronous data characters normally contain 8 data bits (including the parity bit) plus a start bit and at least 1 stop bit, giving a total of 10 bits.
4. Data can be transmitted in blocks of characters known as *transmission blocks*.
5. Asynchronous transmission is only really suitable for relatively low data rates (up to 3 Kbits).
6. Many of the bits transmitted in each block are control bits, giving a **high proportion of overhead.**
7. It is used mainly for applications where character data is generated at irregular intervals (e.g. user input from a keyboard).

**Synchronous Transmission**

1. With synchronous transmission, the receiver's clock is synchronized with the transmitter's clock.
2. Data is transmitted in a continuous stream, and the arrival time of each can be predicted by the receiver.
3. This is achieved either by using a separate timing circuit, or by embedding the timing information in the signal itself. The can be achieved using bi-phase encoding (e.g. Manchester encoding).
4. To keep the transmitter and receiver synchronized, data is transmitted in frames. Frame includes special instruction and control bit for synchronization
5. Synchronous transmission uses no start and stop bits but instead synchronises transmission speeds at both the receiving and sending end of the transmission using clock signals built into each component.
6. Due to there being no start and stop bits the data transfer rate is quicker



1. For large blocks of data, synchronous transmission is far more efficient than asynchronous transmission, requiring far less overhead.
2. The accuracy of the timing information allows much higher data rates.