Exercise 1: Is the "Transmit Data" (TxD) signal an input or an output? How about "Receive Data" (RxD)? Is a computer a 'modem' or a 'terminal'?

- it depends < DTE – TxD output
  DCE – TxD input.

DTE- TxD input
DCE RxD o/p.

- most PCs are used as terminals & wired as DTEs

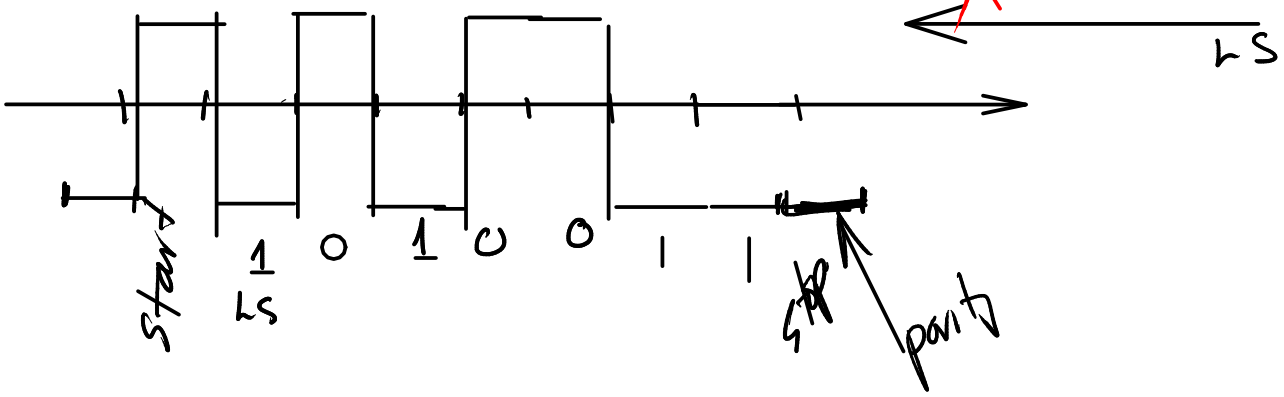
- some computers or communication equipment is wired as DCE

Exercise 2: Draw the waveform used to send the ASCII character 'e' (hex 65) at 9600 bps with seven data bits and no parity.

0x65

~~0~~1100101

LS



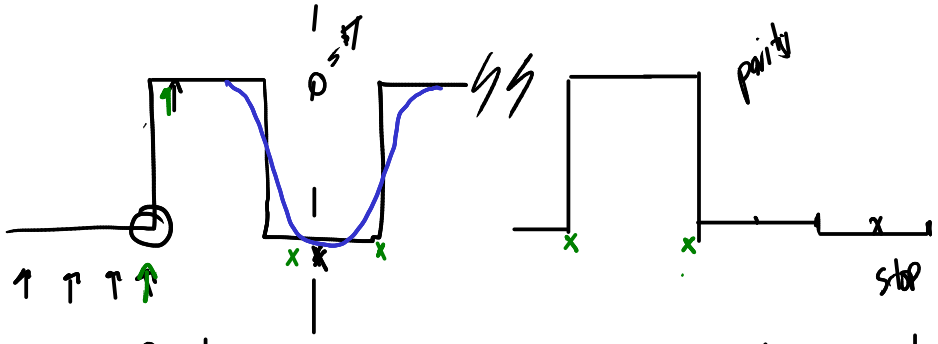
Exercise 3: Will the parity bit allow the receiver to detect all single-bit errors? All double-bit errors?

e.g 0x61 : 3 '1' bits for odd parity: parity = 0
even parity: parity = 1

YES: all single-bit errors detected. (all odd # of errors)

NO: won't detect even number of errors (2, 4...)

Exercise 4: What happens if the receiver's clock is running faster than the transmitter clock?



if RX clock faster → samples earlier & earlier for each bit

↳ if not low

possibly: make error by sampling wrong bit ⇒ framing error

potentially detected as parity or framing error.

Exercise 5: What would happen if the receiver was expecting 8-bit characters and the transmitter was sending 7-bit characters? What about the reverse case?

Tx → S 0 1 2 3 4 5 6 S — low = ← Rx

Rx → S 0 1 2 3 4 5 6 7 S ← Tx

bit = 1

receiver will see MS bit set on all characters.

if opposite case: if bit 7 (MS bit) is '1' → L → extra stop bit

if bit 7 (MS bit) is '0' → H → framing error.