SERIAL Interface

- To calculate the checksum byte of a series of bytes of data
  - Add the bytes together and drop the carries
  - Take the 2's complement of the total sum, and it becomes the last byte of the series
- To perform the checksum operation, add all the bytes, including the checksum byte
  - The result must be zero
  - If it is not zero, one or more bytes of data have been changed

AND ASCII
PLICATION
ROGRAMS

ksum Byte
n ROM
(cont')

Assume that we have 4 bytes of hexadecimal data: 25H, 62H, 3FH, and 52H.(a) Find the checksum byte, (b) perform the checksum operation to ensure data integrity, and (c) if the second byte 62H has been changed to 22H, show how checksum detects the error.

Solution:

(a) Find the checksum byte.

```
        25H
  +     62H
  +     3FH
  +     52H
       118H
```

The checksum is calculated by first adding the bytes. The sum is 118H, and dropping the carry, we get 18H. The checksum byte is the 2's complement of 18H, which is E8H

(b) Perform the checksum operation to ensure data integrity.

```
        25H
  +     62H
  +     3FH
  +     52H
  +     E8H
       200H (dropping the carries)
```

Adding the series of bytes including the checksum byte must result in zero. This indicates that all the bytes are unchanged and no byte is corrupted.

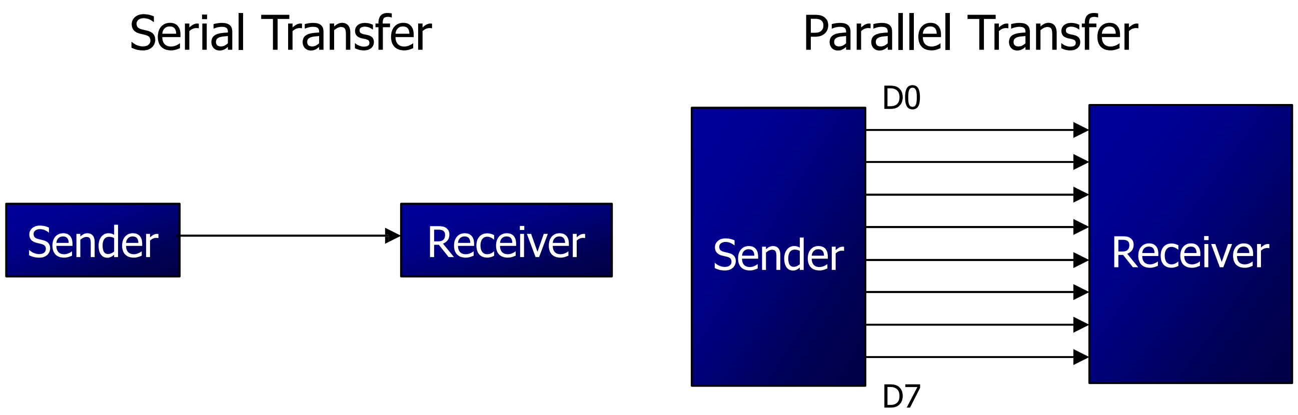
(c) If the second byte 62H has been changed to 22H, show how checksum detects the error.

```
        25H
  +     22H
  +     3FH
  +     52H
  +     E8H
       1C0H (dropping the carry, we get C0H)
```

Adding the series of bytes including the checksum byte shows that the result is not zero, which indicates that one or more bytes have been corrupted.

BASICS OF SERIAL COMMUNICATION

- Computers transfer data in two ways:
  - Parallel
    - Often 8 or more lines (wire conductors) are used to transfer data to a device that is only a few feet away
  - Serial
    - To transfer to a device located many meters away, the serial method is used
    - The data is sent one bit at a time



BASICS OF SERIAL COMMUNICA-TION (cont')

- At the transmitting end, the byte of data must be converted to serial bits using parallel-in-serial-out shift register
- At the receiving end, there is a serial-in-parallel-out shift register to receive the serial data and pack them into byte
- When the distance is short, the digital signal can be transferred as it is on a simple wire and requires no modulation
- If data is to be transferred on the telephone line, it must be converted from 0s and 1s to audio tones
  - This conversion is performed by a device called a modem, "Modulator/demodulator"

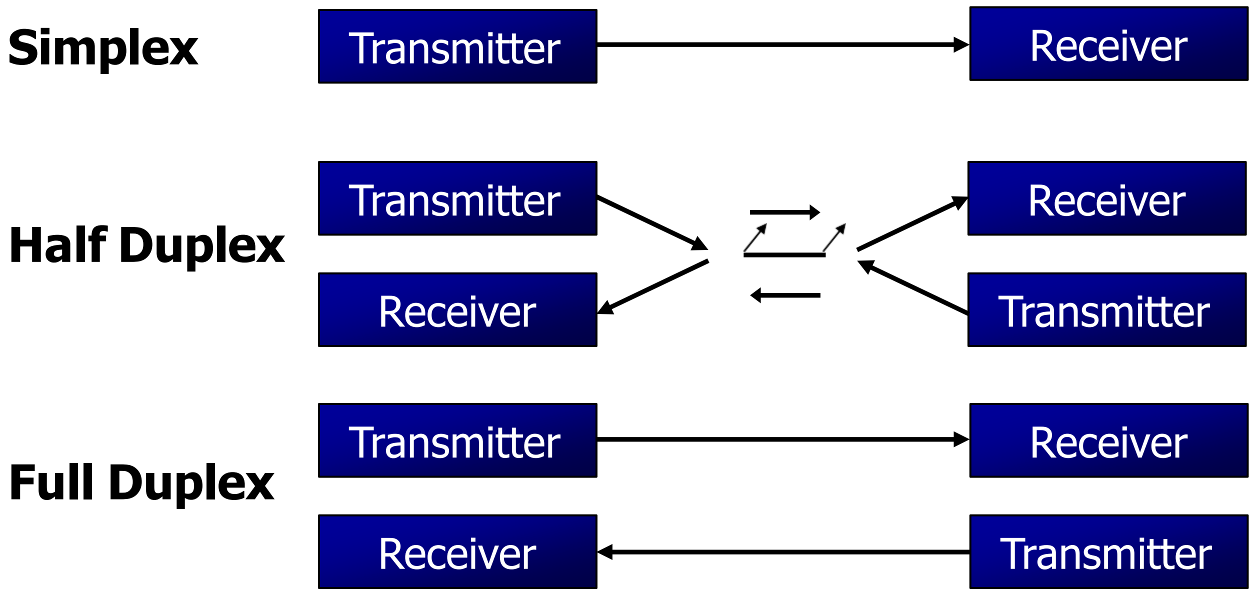
BASICS OF SERIAL COMMUNICA-TION (cont')

- Serial data communication uses two methods
  - Synchronous method transfers a block of data at a time
  - Asynchronous method transfers a single byte at a time
- It is possible to write software to use either of these methods, but the programs can be tedious and long
  - There are special IC chips made by many manufacturers for serial communications
    - UART (universal asynchronous Receiver-transmitter)
    - USART (universal synchronous-asynchronous Receiver-transmitter)

BASICS OF SERIAL COMMUNICA-TION

Half- and Full-Duplex Transmission

- If data can be transmitted and received, it is a duplex transmission
  - If data transmitted one way a time, it is referred to as half duplex
  - If data can go both ways at a time, it is full duplex
- This is contrast to simplex transmission



BASICS OF SERIAL COMMUNICA-TION

Start and Stop Bits

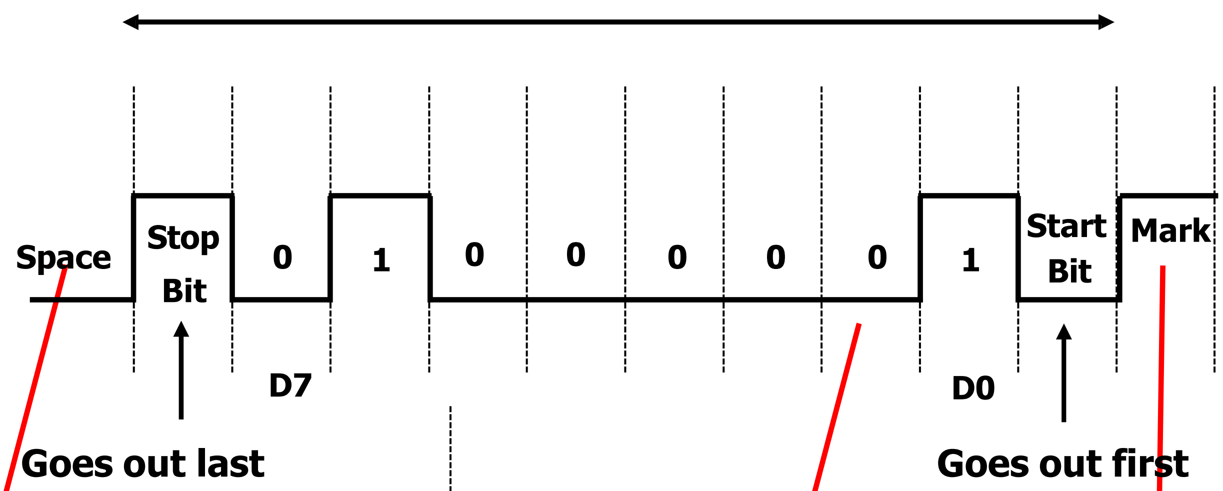
- A protocol is a set of rules agreed by both the sender and receiver on
  - How the data is packed
  - How many bits constitute a character
  - When the data begins and ends
- Asynchronous serial data communication is widely used for character-oriented transmissions
  - Each character is placed in between start and stop bits, this is called framing
  - Block-oriented data transfers use the synchronous method
- The start bit is always one bit, but the stop bit can be one or two bits

BASICS OF SERIAL COMMUNICATION

Start and Stop Bits (cont')

- The start bit is always a 0 (low) and the stop bit(s) is 1 (high)

ASCII character "A" (8-bit binary 0100 0001)



The 0 (low) is referred to as *space*

The transmission begins with a start bit followed by D0, the LSB, then the rest of the bits until MSB (D7), and finally, the one stop bit indicating the end of the character

When there is no transfer, the signal is 1 (high), which is referred to as *mark*

BASICS OF SERIAL COMMUNICA-TION

Start and Stop Bits (cont')

- Due to the extended ASCII characters, 8-bit ASCII data is common
  - In older systems, ASCII characters were 7-bit
- In modern PCs the use of one stop bit is standard
  - In older systems, due to the slowness of the receiving mechanical device, two stop bits were used to give the device sufficient time to organize itself before transmission of the next byte

BASICS OF SERIAL COMMUNICA-TION

Start and Stop Bits (cont')

- Assuming that we are transferring a text file of ASCII characters using 1 stop bit, we have a total of 10 bits for each character
  - This gives 25% overhead, i.e. each 8-bit character with an extra 2 bits
- In some systems in order to maintain data integrity, the parity bit of the character byte is included in the data frame
  - UART chips allow programming of the parity bit for odd-, even-, and no-parity options

BASICS OF SERIAL COMMUNICA-TION

Data Transfer Rate

- The rate of data transfer in serial data communication is stated in bps (bits per second)
- Another widely used terminology for bps is baud rate
  - It is modem terminology and is defined as the number of signal changes per second
  - In modems, there are occasions when a single change of signal transfers several bits of data
- As far as the conductor wire is concerned, the baud rate and bps are the same, and we use the terms interchangeably

BASICS OF SERIAL COMMUNICA-TION

Data Transfer Rate (cont')

- The data transfer rate of given computer system depends on communication ports incorporated into that system
  - IBM PC/XT could transfer data at the rate of 100 to 9600 bps
  - Pentium-based PCs transfer data at rates as high as 56K bps
  - In asynchronous serial data communication, the baud rate is limited to 100K bps

BASICS OF SERIAL COMMUNICA-TION

RS232 Standards

- An interfacing standard RS232 was set by the Electronics Industries Association (EIA) in 1960
- The standard was set long before the advent of the TTL logic family, its input and output voltage levels are not TTL compatible
  - In RS232, a 1 is represented by -3 ~ -25 V, while a 0 bit is +3 ~ +25 V, making -3 to +3 undefined

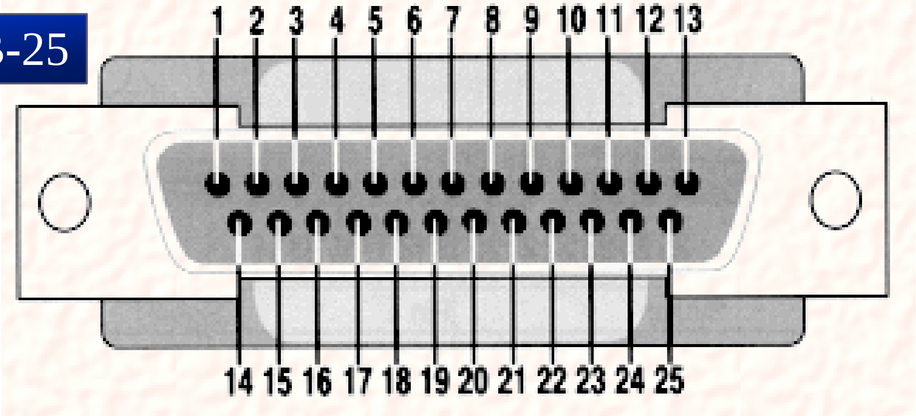
BASICS OF SERIAL COMMUNICA-TION

RS232 Standards (cont')

RS232 DB-25 Pins

Pin	Description	Pin	Description
1	Protective ground	14	Secondary transmitted data
2	Transmitted data (TxD)	15	Transmitted signal element timing
3	Received data (RxD)	16	Secondary receive data
4	Request to send (-RTS)	17	Receive signal element timing
5	Clear to send (-CTS)	18	Unassigned
6	Data set ready (-DSR)	19	Secondary receive data
7	Signal ground (GND)	20	Data terminal ready (-DTR)
8	Data carrier detect (-DCD)	21	Signal quality detector
9/10	Reserved for data testing	22	Ring indicator (RI)
11	Unassigned	23	Data signal rate select
12	Secondary data carrier detect	24	Transmit signal element timing
13	Secondary clear to send	25	Unassigned

RS232 Connector DB-25

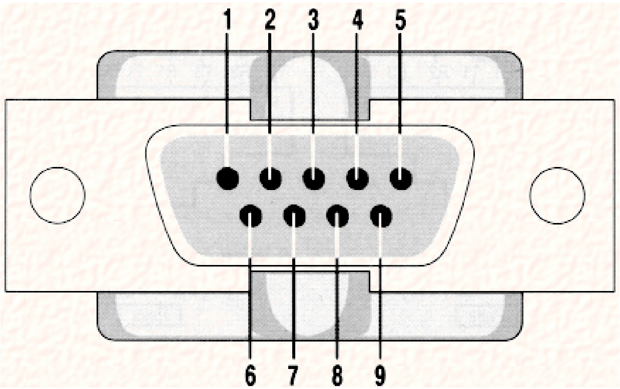


BASICS OF SERIAL COMMUNICATION

RS232 Standards (cont')

- Since not all pins are used in PC cables, IBM introduced the DB-9 version of the serial I/O standard

RS232 Connector DB-9



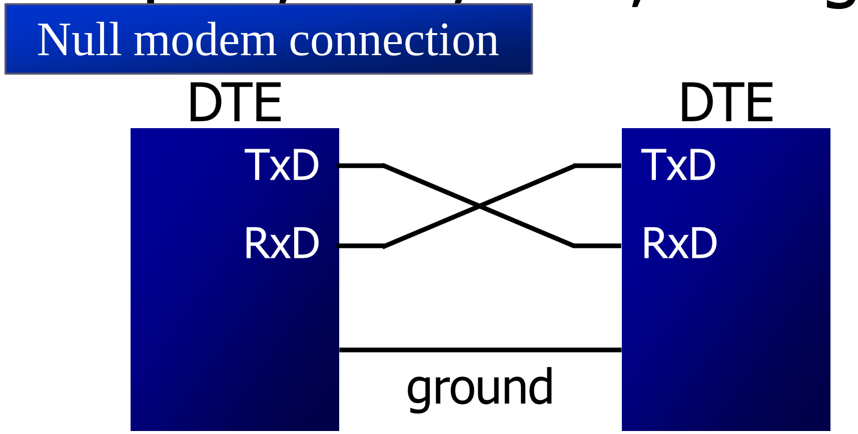
RS232 DB-9 Pins

Pin	Description
1	Data carrier detect (-DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (-DSR)
7	Request to send (-RTS)
8	Clear to send (-CTS)
9	Ring indicator (RI)

BASICS OF SERIAL COMMUNICA-TION

Data Communication Classification

- Current terminology classifies data communication equipment as
  - DTE (data terminal equipment) refers to terminal and computers that send and receive data
  - DCE (data communication equipment) refers to communication equipment, such as modems
- The simplest connection between a PC and microcontroller requires a minimum of three pins, TxD, RxD, and ground



BASICS OF SERIAL COMMUNICATION

RS232 Pins

- DTR (data terminal ready)
  - When terminal is turned on, it sends out signal DTR to indicate that it is ready for communication
- DSR (data set ready)
  - When DCE is turned on and has gone through the self-test, it assert DSR to indicate that it is ready to communicate
- RTS (request to send)
  - When the DTE device has byte to transmit, it assert RTS to signal the modem that it has a byte of data to transmit
- CTS (clear to send)
  - When the modem has room for storing the data it is to receive, it sends out signal CTS to DTE to indicate that it can receive the data now

BASICS OF SERIAL COMMUNICATION

RS232 Pins (cont')

- DCD (data carrier detect)
  - The modem asserts signal DCD to inform the DTE that a valid carrier has been detected and that contact between it and the other modem is established
- RI (ring indicator)
  - An output from the modem and an input to a PC indicates that the telephone is ringing
  - It goes on and off in synchronous with the ringing sound

8051 CONNECTION TO RS232

- A line driver such as the MAX232 chip is required to convert RS232 voltage levels to TTL levels, and vice versa
- 8051 has two pins that are used specifically for transferring and receiving data serially
  - These two pins are called TxD and RxD and are part of the port 3 group (P3.0 and P3.1)
  - These pins are TTL compatible; therefore, they require a line driver to make them RS232 compatible