EE23DP001
Aditya Shirodkar

EE690: Embedded Systems Design
Homework 1

Q1) Distinguish between UART, USART, RS-232, and RS-485, and also mention the physical layer specifications for each (voltage levels, timing, frequencies, wiring conventions, etc).

A1)

UART:

- Description:
    - UART is an abbreviation for "Universal asynchronous receiver/transmitter".
    - It is a large integration device that is designed to deal with the transmission of sequential data.
- Voltage Levels:
    - Uses TTL voltage levels
    - 0v for logic 0 and 5v for logic 1.
- Timings:
    - Uses Baud Rates
    - Typical baud rates are 9600, 19200, 38400, and 115200 bps
- Frequency:
    - UART interfaces have a maximum data rate of around 5 Mbps.
    - The receive UART uses a clock that is 16 times the data rate.

- Wiring Convention:
    - UART uses 2 wires
        - Transmitter (Tx)
        - Receiver (Rx)

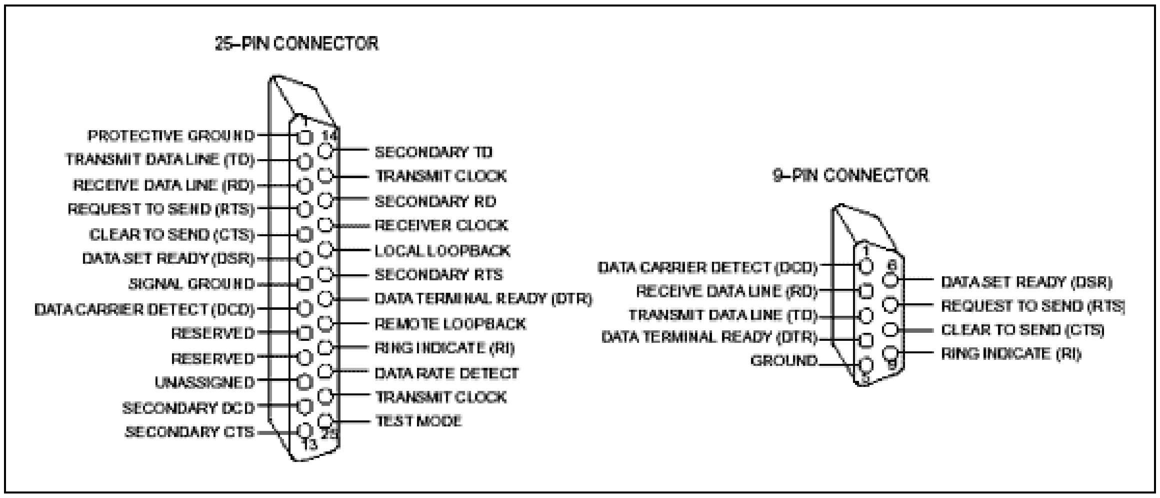
USART:

- Description:
    - A USART (universal synchronous/asynchronous receiver/transmitter) is hardware that enables a device to communicate using serial protocols.
    - It can function in a slower asynchronous mode, like a universal asynchronous receiver/transmitter (UART), or in a faster synchronous mode with a clock signal.
- Voltage Levels:
    - Uses TTL voltage levels
    - 0v for logic 0 and 5v for logic 1.

- Timings:
    - Uses baud rate
    - Typical baud rates are 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200
- Frequency:
    - USART can achieve speeds upto 115.2 kbit/s
- Wiring Convention:
    - USART uses Tx (Transmit), Rx(Receive), and GND wires.

RS-232:

- Description:
    - RS-232 or Recommended Standard 232 is a standard for serial communication transmission of data. It formally defines signals connecting between a DTE (data terminal equipment) such as a computer terminal, and a DCE (data circuit-terminating equipment or data communication equipment), such as a modem.
    - It is used for serial communication, it is used for connecting computer and its peripheral devices to allow serial data exchange between them.
- Voltage Levels:
    - Data 1: -5V (min) -15V (max)
    - Data 0: 5V (min) 15V (max)
- Timings:
    - Common baud rates used are 2.4k, 9.6k, 19.2k
- Frequency:
    - RS-232 can operate on different bit rates, standard values lie between 110 bit/s and 115200 bit/s.
- Wiring Convention:
    - RS-232 specifies a 25-pin connector as the minimum connector size that can accommodate all the signals defined in the functional portion of the standard.



RS-485:

- Description:
  - RS-485, also known as TIA-485(-A) or EIA-485, is a standard defining the electrical characteristics of drivers and receivers for use in serial communications systems.
- Voltage Levels:
  - High/1: Voltage > 200 mV
  - Low/0: Voltage < -200 mV
- Timings:
  - RS-485 has standard baud rates up to 115200 baud
- Frequency:
  - RS-485 permits up to 10Mbps data rates
- Wiring Convention:
  - 9 pins; using DB-9 connector

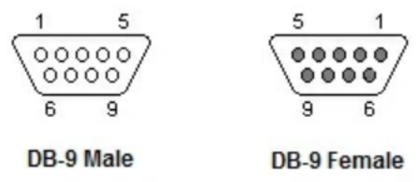
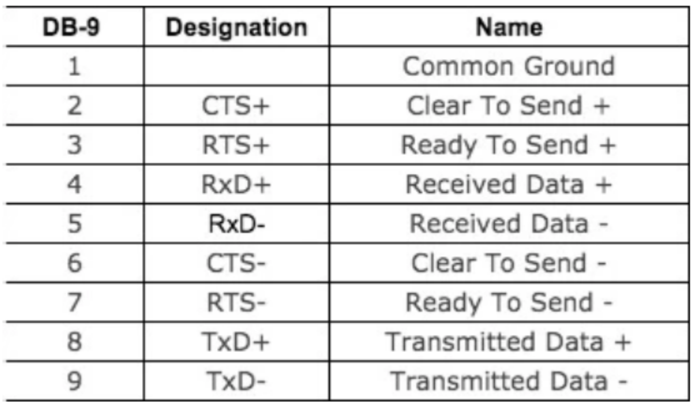


Figure 1. RS485 connector pinout



DB-9	Designation	Name
1		Common Ground
2	CTS+	Clear To Send +
3	RTS+	Ready To Send +
4	RxD+	Received Data +
5	RxD-	Received Data -
6	CTS-	Clear To Send -
7	RTS-	Ready To Send -
8	TxD+	Transmitted Data +
9	TxD-	Transmitted Data -

Figure 2. RS485 pinout

Q2) What is flow control in serial communications ?

A2) Serial devices, such as printers and modems, do not process data as quickly or efficiently as the computers they are connected to. The term flow control is used to describe the method in which a serial device controls the amount of data being transmitted to itself.

UART Flow Control is a method for slow and fast devices to communicate with each other over UART without the risk of losing data.

a) **Hardware Flow Control:**
With hardware flow control (also called RTS/CTS flow control), two extra wires are needed in addition to the data lines. They are called RTS (Request to Send) and CTS (Clear to Send). These wires are cross-coupled between the two devices, so RTS on one device is connected to CTS on the remote device and vice versa. Each device will use its RTS to output if it is ready to accept new data and read CTS to see if it is allowed to send data to the other device.

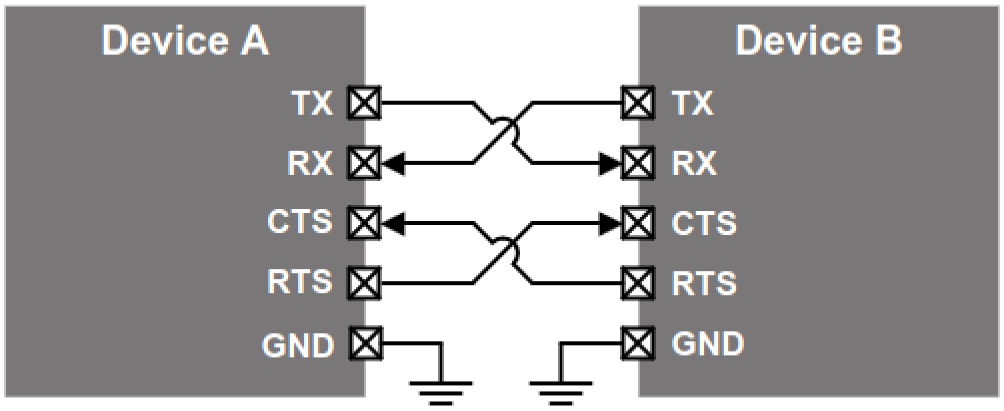


Figure 3.1. Hardware Flow Control

Software Flow Control:
Software flow control does not use extra wires. Only 3 wires are required (RX, TX, and GND). Transmission is started and stopped by sending special flow control characters. The flow control characters are sent over the normal TX and RX lines. The flow control characters are typically the ASCII codes XON and XOFF (0x11 and 0x13). If device A sends XOFF to device B it means that B should halt transmission to A until B receives an XON character from A

Q3) Normally a DB-9 connector is used for serial communication. What does DB-9 stand for? What are each of the pins used for? Why are 9 pins needed when serial only needs two lines (TX and RX)?

A3) The DB9 connector (originally named DE-9) is an analog socket, with 9 pins, from the D-Subminiatures (D-Sub) connector family. The DB9 has the smallest "footprint" of the D-Subminiature connectors. The prefix “D” represents the D-shape of the connector shell. The DB9 connector is mainly used in serial ports, allowing asynchronous data transmission according to the RS-232 standard (RS-232C).

DB9 connectors are designed to work with the EIA/TIA 232 serial interface standard, which determined the function of all nine pins as a standard, so that multiple companies could design them into their products. DB9 connectors were commonly used for serial peripheral devices like keyboards, mice, joysticks, etc.

DB-9 Connector pin functions:

Pin	**Function**
1	DCD (Data Carier Detect)
2	RXD (Received Datta)
3	TXD (Transmitted Data)
4	DTR (Data Terminal Ready)
5	Ground
6	DSR (Data Set Ready)
7	RTS (Request To Send)
8	CTS (Clear To Send)
9	RI (Ring Indicator)

Q4) A GPS reveiver chip communicates via UART at 115200 baud. When powered "ON", it sends out a string of data that contains the module status as well as the current GPS location and GPS derived time (an example of the string is given here: https://www.gpsworld.com/what-exactly-is-gps-nmea-data). Write (pseudocode) for a program that:
-reads incoming serial data from the module.
-when a GLL string is received (see string definition here on p11 https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual-Rev2.1-Dec07.pdf if checksum is valid and status is valid, then the time portion of the string should be sent out over the UART TX pin to another device

A4)

Consider a sample message:
$GPGGA,181908.00,3404.7041778,N,07044.3966270,W,4,13,1.00,495.144,M,29.200,M,0.10,0000*40

Observations:

- All parameters sent are comma separated
- All parameters can be saved as character strings
- An array of character strings can be used
- Checksum can be converter to char form and compared

Pseudocode:

- Configure UART for Transmission and Reception on interrupt, with 115200 baud and FIFO enabled
- (Refer https://www.gpsworld.com/what-exactly-is-gps-nmea-data)
  Define parameters as necessary, and create a structure containing the same
    - char GPS_pos
    - float time
    - float lat_num
    - char lat_dir
    - float lon_num
    - char lon_dir
    - int q_ind
    - int num_sat
    - float hdop
    - float alt
    - char alt_unit
    - float geoidal_sep

    - char geoidal_sep_unit
    - float age_corr
    - int corr_stat_id
    - int checksum

- Define data structure as follows:
  [GPS_pos time lat_num lat_dir lon_num lon_dir q_ind num_sat hdop alt alt_unit geoidal_sep geoidal_sep_unit age_corr corr_stat_id checksum]
- On UART reception,
    - Keep receiving data as long as “,” is not received
    - If “,” is received,
        - Store data received till now in 1st element of data structure
        - Start storing succeeding data in next element of data structure
        - Continue till reception is complete
- Perform checksum on each element of the structure, and perform consecutive XORing of all elements to form final checksum
- Compare with received checksum
  If Calculated_Checksum == checksum
    - Send time on Tx pin of UART