Serial Communication

Prof. James L. Frankel
Harvard University

Version of 11:20 PM 1-Dec-2021

Overview of the Serial Protocol

- Simple protocol for communicating data – usually character by character
  - Existed before more complex protocols such as Ethernet, USB, etc.
- Data is transmitted and received in a bit serial manner
- Transmission and reception can both proceed at the same time
  - For serial communication, this is referred to as “full duplex” operation
  - Communication that can occur in only one direction is referred to as “simplex” operation
  - If no concurrent transmission and reception is allowed but both directions can occur, this is referred to as “half duplex” operation
- The simplest bi-directional version has three wires in the cable
  - **GND** – A **common ground** wire (for ground reference)
  - **TxD** – A transmit wire
  - **RxD** – A receive wire
- There are many other control wires
- Standardized versions are referred to as
  - RS-232
  - EIA RS-232 or EIA-232
  - TIA-232

Voltage Levels

- Two states for data (& control signals)
  - Space
    - Indicates a **0** for *data*
    - +3 to +15 VDC
      - Originally +12 VDC
    - Indicates **asserted** or **on** or **true** for *control lines*
  - Mark
    - Indicates a **1** for *data*
    - -15 to -3 VDC
      - Originally -12 VDC
    - Indicates **deasserted** or **off** or **false** for *control lines*
- These voltages cause problems
  - Additional power supplies or voltage converters are required
  - Higher voltage limits transmission speed
  - Reference to a transmitted common ground limits cable length and can cause ground loop issues

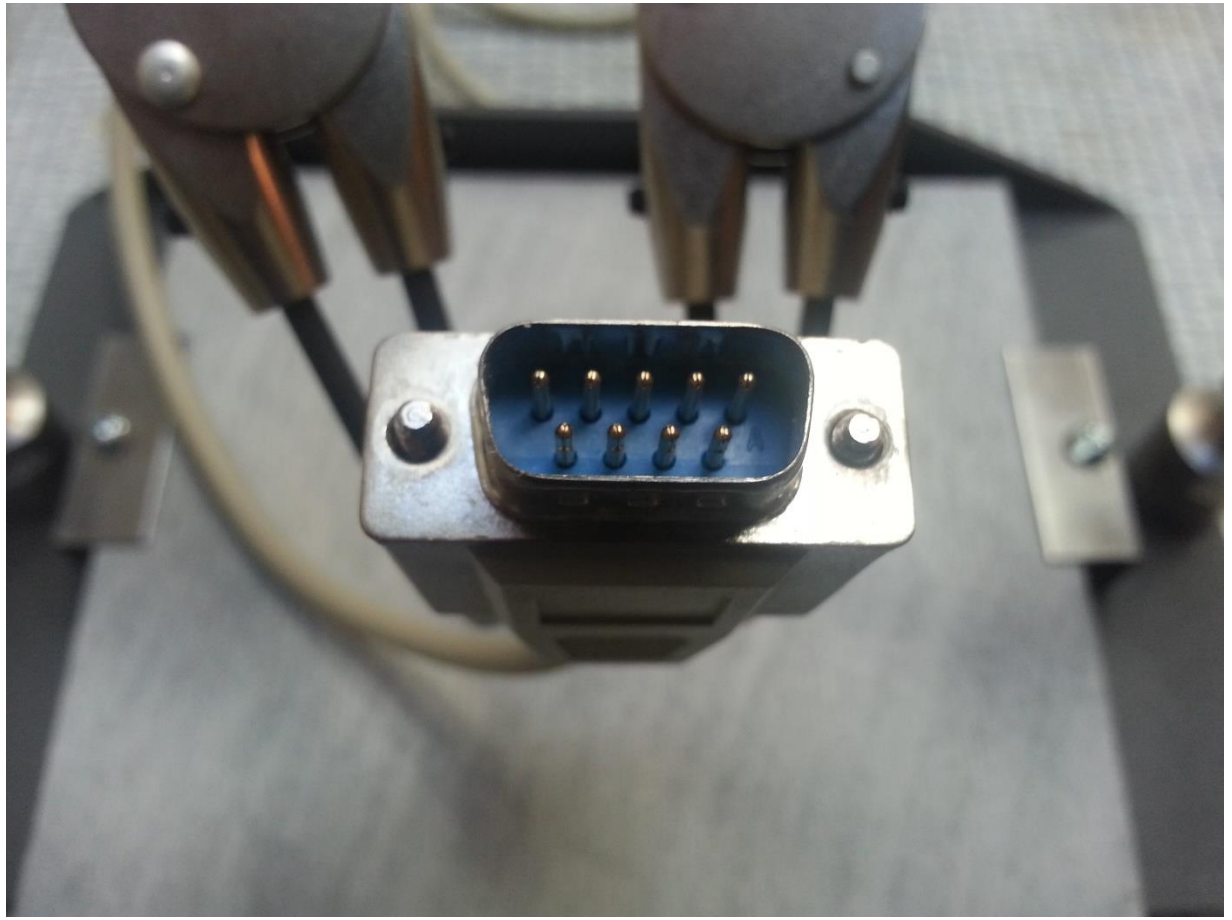
Two Sides of the Connection

- **DTE** – Data Terminal Equipment
  - Examples are terminals & peripheral devices
- **DCE** – Data Communication Equipment
  - Examples are modems & computers
- TxD sends data from DTE to DCE
- RxD sends data from DCE to DTE

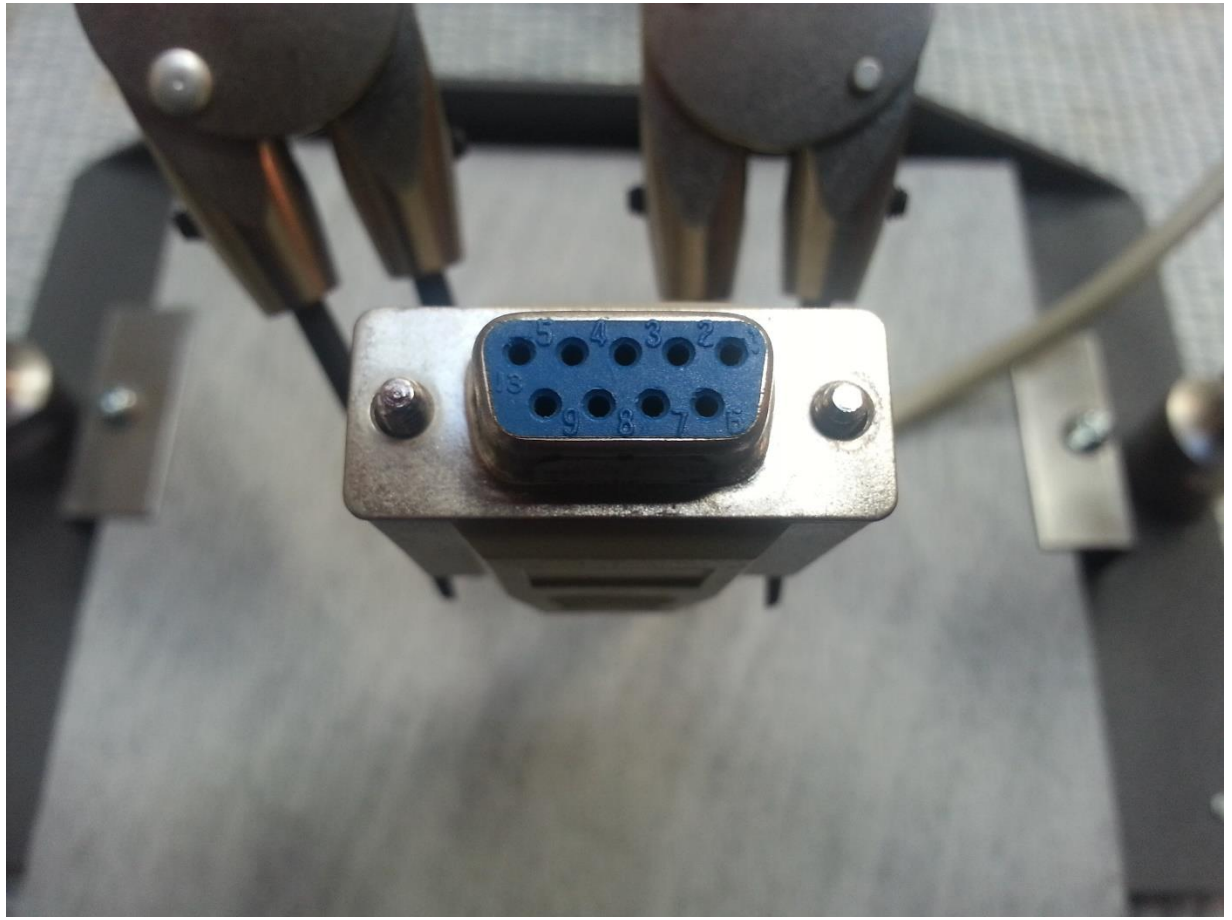
Connectors

- **D-Subminiature Connector** with 25 pins
  - **DB-25**
- D-Subminiature Connector with 9 pins
  - **DE-9** (Often referred to as DB-9)
- A metal shield provides protection from interference
- Connector with pins is referred to as male (DB-25M, DE-9M)
  - DTE defines the pins
- Connector with sockets is referred to as female (DB-25F, DE-9F)
  - DCE defines the pins
- The standard defines which pins are used for which signals

DE-9M Connector



DE-9F Connector



Speed of Connection

- Bit rate (frequency) is named the **baud rate**
- For example, 9,600 baud is 9,600 bits per second
- The baud rate determines the timing per bit (cycle time) – the **bit time**
  - bit time = 1/(baud rate)
  - At 9,600 baud, the bit time is ~0.10417 ms

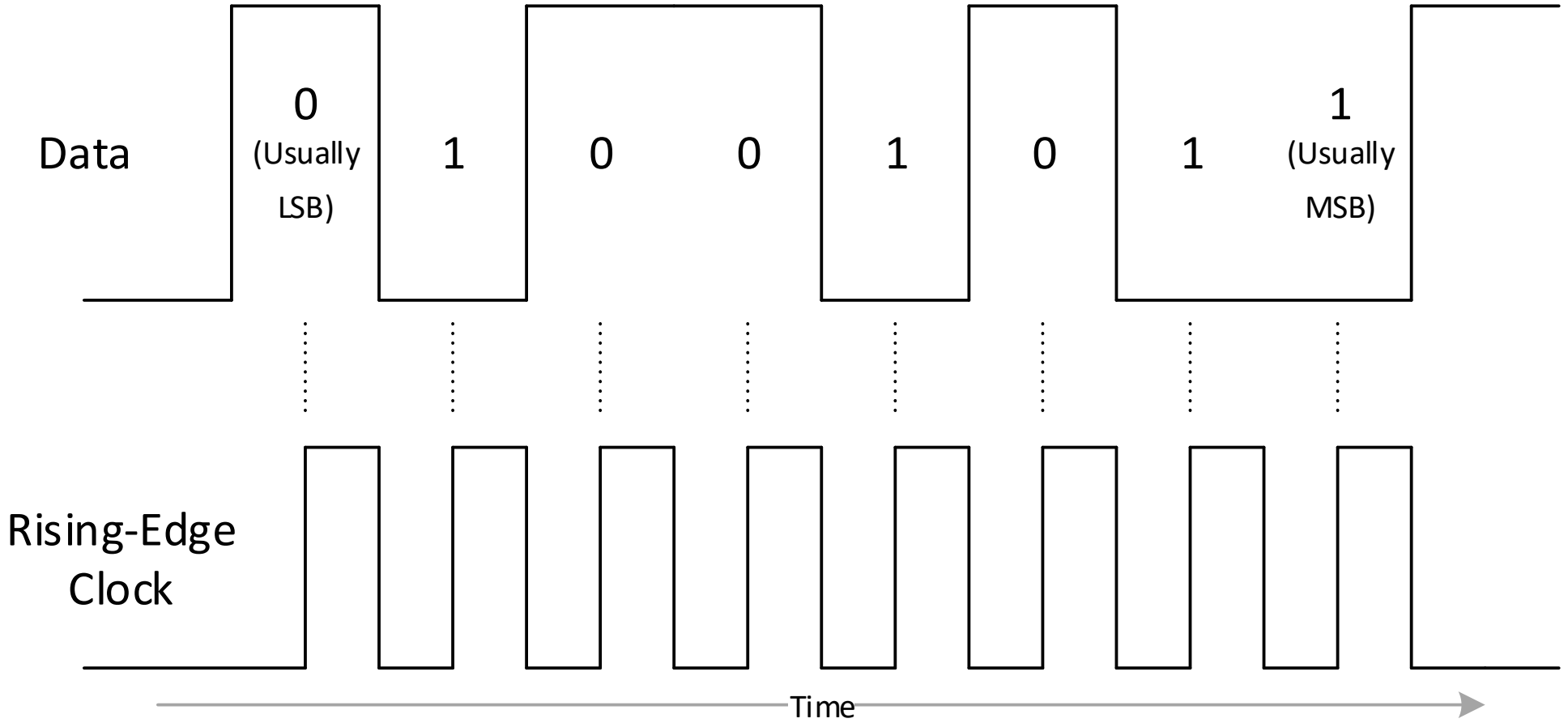
Synchronous vs. Asynchronous

- Synchronous communication transmits a clock with data
- Asynchronous communication embeds an implicit clock in the data
- EIA-232 is asynchronous

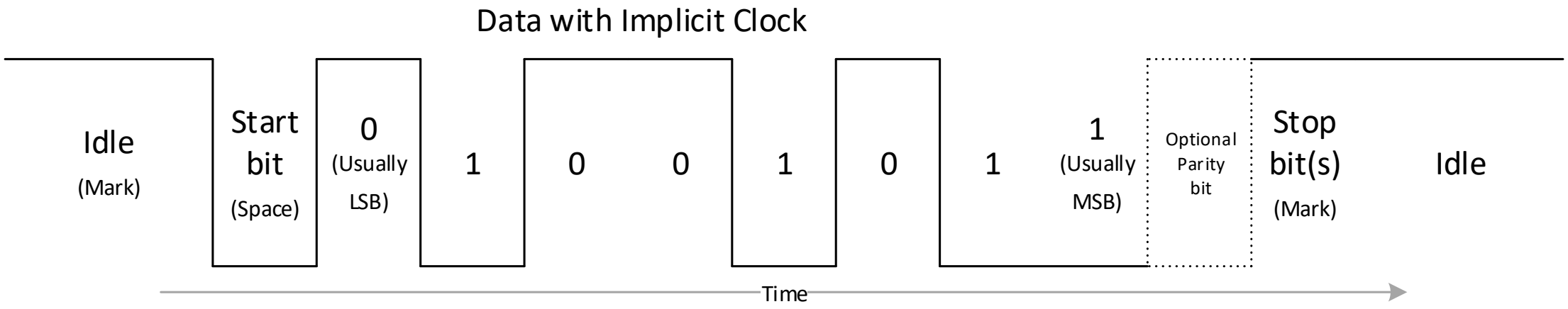
- The two subsequent examples show eight data bits

- All data transmitted together as a character with any necessary signals to allow correct reception is referred to as a **frame**

Example of Synchronous Communication



Example of Asynchronous Communication



Details of Asynchronous Protocol

- To transmit back-to-back data, there is no need for any “idle” period
- Transmitter and receiver must agree on the protocol
  - They must agree on the baud rate, the number of data bits, whether parity is present and, if it is present, whether it is even or odd, and the number of stop bits
- The number of **data bits** is usually between 5 and 9, inclusive
- The optional **parity bit** is used to detect single-bit errors
  - Can be even or odd
    - Even or odd parity is computed/checked from the number of 1 bits in the data including the parity bit
  - The parity bit need not be part of the frame
- The **stop bit(s)** is used to “frame” the data
  - In most usage, there may be 1 stop bit, 1.5 stop bits, or 2 stop bits
- The low start bit and high stop bit(s) mean that there are at least two voltage transitions in each frame
- If the data line is held low for longer than one frame length, this is called a **break** condition

Receiver Action

- Receiver samples the data line at a rate higher than the baud rate – this is called the **sampling rate**
  - Say, at sixteen times the baud rate
- When the receiver detects the **start bit**, it waits for half the bit time and resamples the data line
  - If the start bit is not still present, it was a spurious start bit
  - If the start bit is still present, it is a valid start bit
  - The higher the receiver sample rate, the more accurately the middle of the start bit can be determined
- Subsequent bits are sampled at the bit time from the middle of the start bit
  - This allows for some difference between the clock rate of the transmitter and that of the receiver
- More advanced receivers resynchronize their sampling point whenever they see a change in the data line
- After all the data bits and the optional parity bit, the receiver checks for the **stop bit(s)**

Receiver Buffering

- After a complete frame has been received, the serial data is made available to the computer as a parallel byte or word
- In order to allow the receiver to receive the next data frame, at a minimum, most receivers are able to store the data from one frame while receiving the next frame
- More advanced receivers use a FIFO to store the received parallel data
  - In this usage, the term **FIFO** is used to refer to a hardware FIFO (First-in, First-out) queue
- This buffering allows the receiver to function without losing data before the computer reads data from the receiver

Errors Found by the Receiver

- Spurious start bit
- Parity error
  - Received parity does not have expected value
- Framing error
  - The stop bit was not present where expected
- Overrun error
  - The buffering was insufficient (old data was overwritten by new data)

Component to Transmit/Receive Bit Pattern

- **UART** – Universal Asynchronous Receiver/Transmitter
  - Asynchronous only
- **USART** – Universal Synchronous/Asynchronous Receiver/Transmitter
  - Both Synchronous & Asynchronous

- These devices work at circuit logic voltage level (e.g. 3.3 V)
- An additional device is required to produce and receive the EIA-232 voltage levels
  - Often called a RS-232 Voltage Converter or RS-232 Driver/Receiver
  - For example, see the Maxim MAX232A