Embedded System Architecture - CSEN 701

Module 5: *Communication and Networking*

Lecture 10: *Overview of Communication in ES & Serial Communication (1)*

Dr. Eng. Catherine M. Elias

catherine.elias@guc.edu.eg

Lecturer, Computer Science and Engineering,
Faculty of Media Engineering and Technology, German University in Cairo

Outline

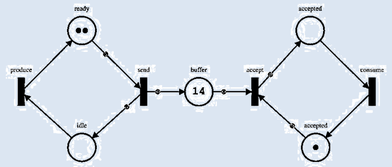
- Basics of Communication
- Communication Protocols
- UART

The big Picture

The Real-time Embedded System

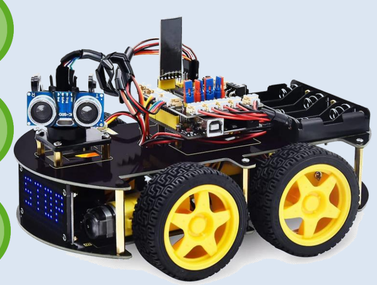
ES Modeling & Design (*2 Modules*)

- System Modeling
- Design Considerations
- Power management and optimization
- Reliability and fault tolerance



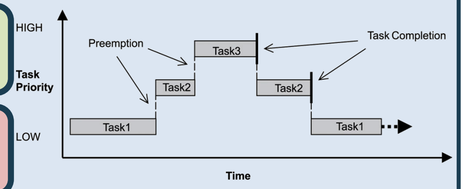
ES Hardware Components (*4 Modules*)

- Microcontroller Fundamentals
- Embedded programming languages
- Embedded Hardware
- Communication and Networking



ES Software Components (*1 Module*)

- Real-Time Systems
- Multi-tasking
- Scheduling
- Resource Management



Embedded System Tools & Software Development (*2 Modules*)

- Debugging techniques
- Interrupts and exception handling
- Memory management

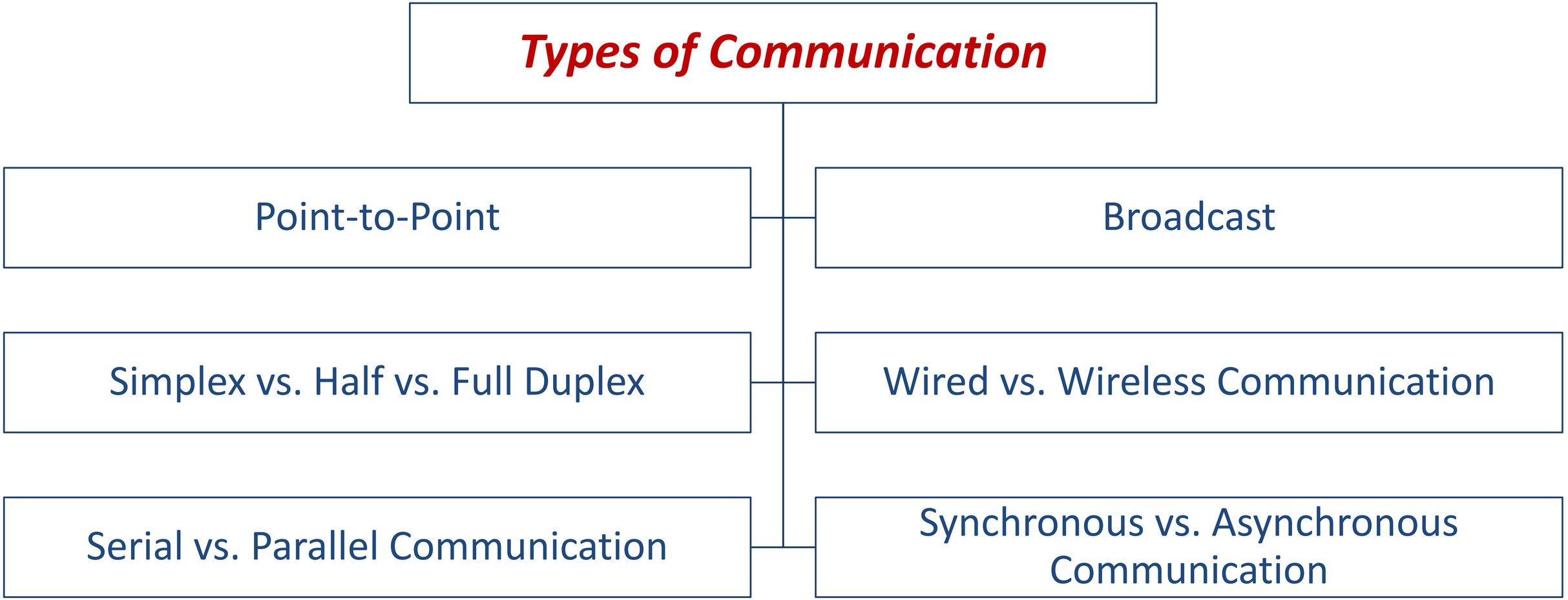
Overview of Communication Systems

- The term "***Communication Systems***" refers to the organized and systematic processes of **transmitting, receiving, and exchanging information** between two or more entities.
- In the context of electronic and technological systems, communication systems play a crucial role in facilitating the transfer of data, signals, or messages from a source to a destination.
- These systems are designed to **enable effective and reliable information exchange in various forms**, such as voice, text, video, or other data formats.

Basic Elements of a Communication System

- **Source:** The origin of the information or data that needs to be communicated.
- **Transmitter:** Converts the information into a form suitable for transmission (signals, codes, etc.).
- **Receiver:** Captures and converts the transmitted information back into a usable form.
- **Channel:** The medium through which the information is transmitted (e.g., cables, airwaves, fiber optics).
- **Destination:** The final point where the information is intended to be received and utilized.

Basics of Communication

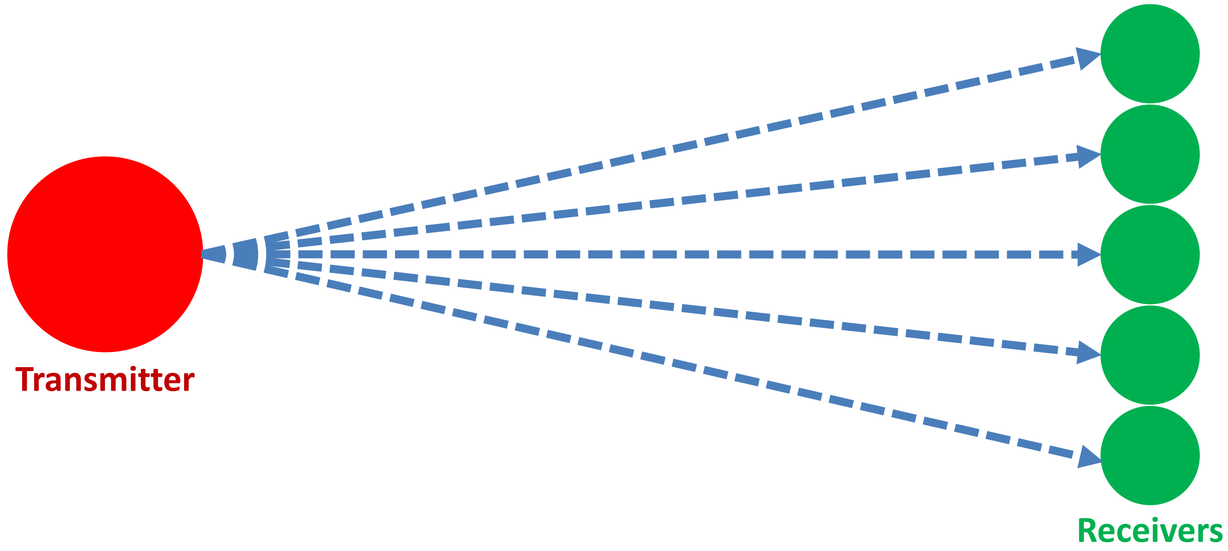


Types of Communication: *Point-to-Point Communication*

- **Definition:** Communication between **two specific entities** or devices. It is a dedicated link between a transmitter and a receiver.
- **Example:** Traditional telephone communication is a classic example of point-to-point communication

Types of Communication: *Broadcast Communication*

- **Definition:** Communication where data is transmitted from **one transmitter to multiple recipients simultaneously**.
- **Example:** Radio and television broadcasting are common examples of broadcast communication.



Types of Communication: *Simplex vs. Half vs. Full Duplex*

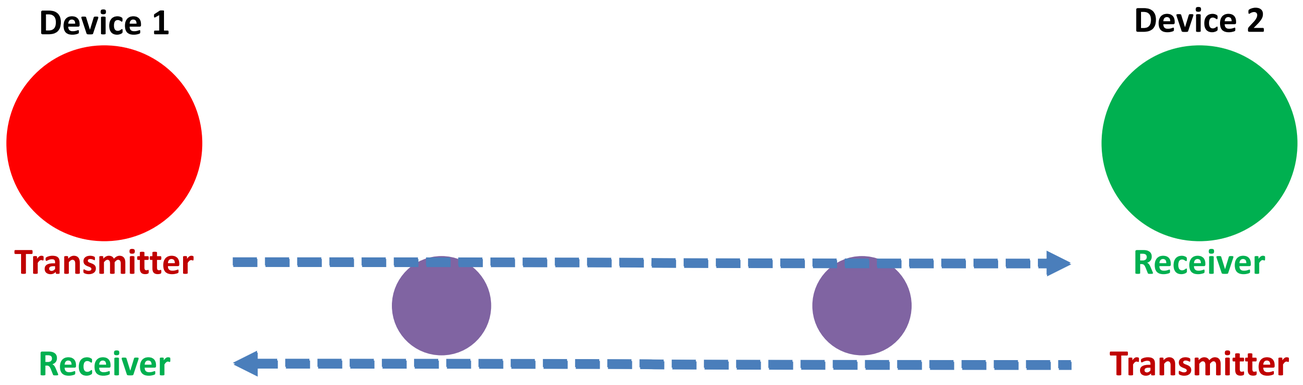
- **Simplex** Communication
  - **Definition:** Unidirectional communication where data flows in only one direction.
  - **Example:** Television broadcasting, where the TV station sends signals to viewers without expecting direct responses.

Types of Communication: *Simplex vs. Half vs. Full Duplex*

- **Full-Duplex Communication**
  - **Definition:** Bidirectional communication where data can be transmitted in both directions simultaneously.
  - **Example:** Traditional telephone conversations, where both parties can speak and listen at the same time.

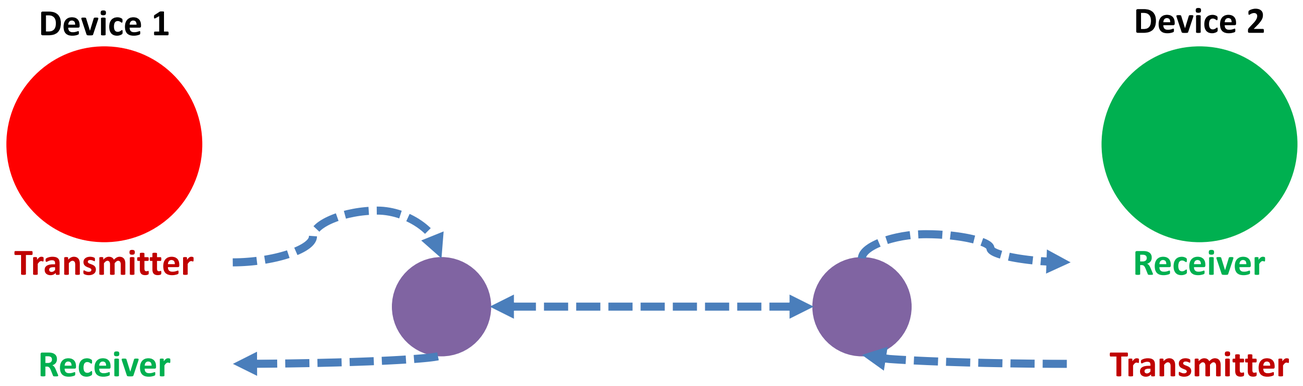
Types of Communication: *Simplex vs. Half vs. Full Duplex*

- **Half-Duplex Communication**
  - **Definition:** Bidirectional communication, but only one direction at a time. Devices can both send and receive, but not simultaneously.
  - **Example:** Walkie-talkies are often half-duplex; while one person talks, the other listens.



Types of Communication: *Simplex vs. Half vs. Full Duplex*

- **Half-Duplex Communication**
  - **Definition:** Bidirectional communication, but only one direction at a time. Devices can both send and receive, but not simultaneously.
  - **Example:** Walkie-talkies are often half-duplex; while one person talks, the other listens.



Types of Communication: *Wired vs. Wireless Communication*

- **Wireless Communication**
  - **Medium:** Utilizes radio frequencies or other wireless means for data transmission.
  - **Examples:** Wi-Fi, Bluetooth, cellular networks.
- **Wired Communication**
  - **Medium:** Uses physical cables or conductors for data transmission.
  - **Examples:** Ethernet cables, fiber optic cables.

Types of Communication: *Synchronous vs. Asynchronous Communication*

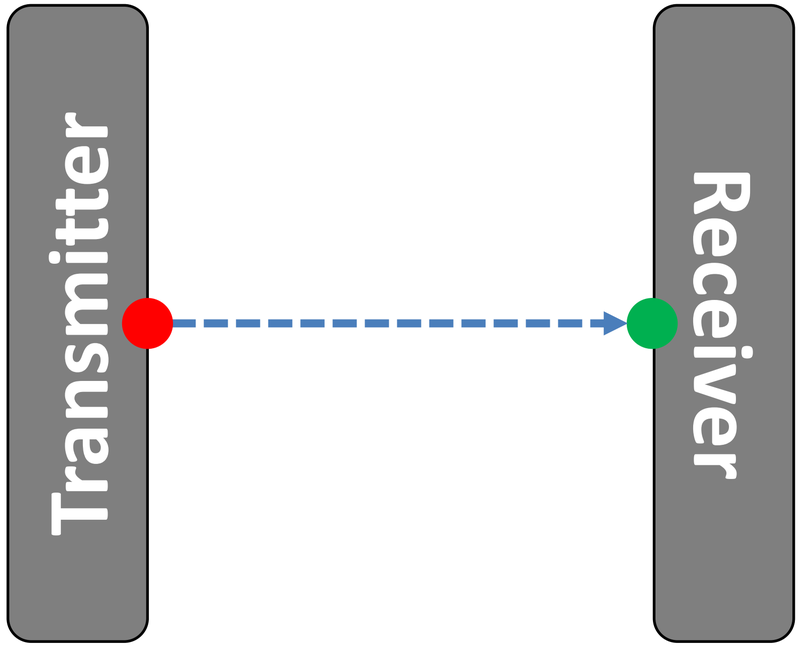
- Coordination between the transmitter and receiver is required
  - when to start the transmission and when to end it,
  - when one particular bit or byte ends and another begins,
  - when the receiver's capacity has been exceeded…

- **Synchronous Communication**
  - **Timing:** transmitter and receiver are synchronized in time.
  - **Wiring:** two different signals, a pulse on one signal line indicates when another bit of information is ready on the other signal line.
  - **Example:** Synchronous serial communication.
  - High Risk for deadlocks

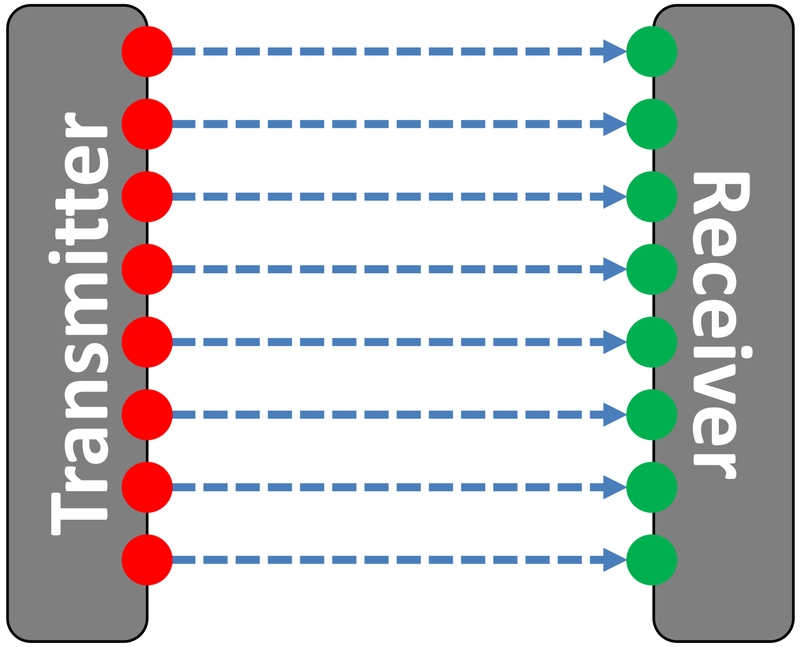
- **Asynchronous Communication**
  - **Timing:** No fixed timing between transmitter and receiver.
  - **Wiring:** only uses one signal
  - **Example:** Asynchronous serial communication, such as UART.

Types of Communication: *Serial vs. Parallel Communication*

- **Serial Communication**
  - **Data Transmission**: Sends data bit by bit over a single communication line.
  - **Example:** RS-232 serial communication.



- **Parallel Communication**
  - **Data Transmission:** Sends multiple bits of data simultaneously over multiple communication lines.
  - **Example:** Parallel ports in computers.



Types of Communication: *Serial vs. Parallel Communication*

Serial Communication	Parallel Communication
one bit at a time	one byte/word at a time
sync or async	sync
no data skew	data skew
noise immunity	crosstalk
simplex, half, or full duplex	simplex or half duplex
complex design	simple design
two/four wires	n-wires

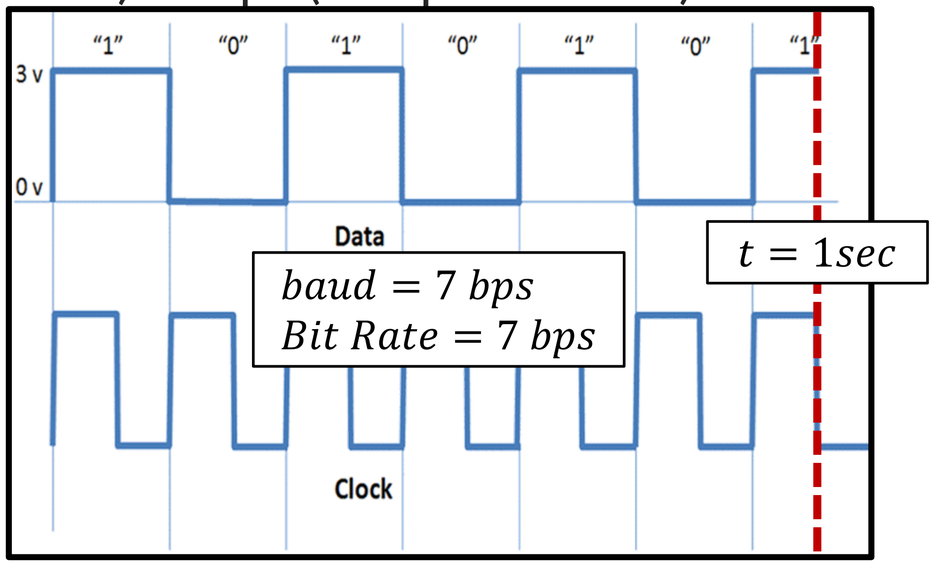
Basics of Communication

Communication Systems Parameters

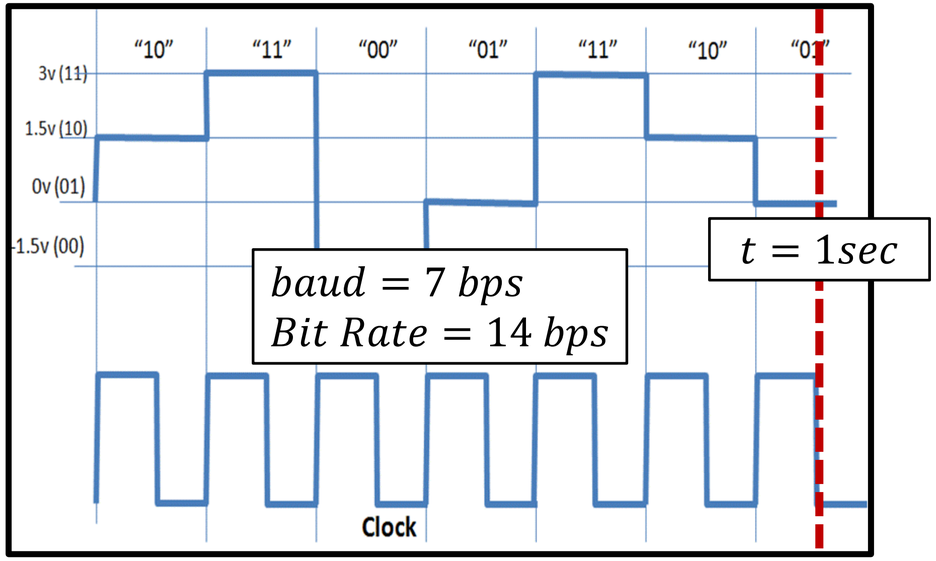
- Baud Rate
- Bit Rate (Data Rate)
- Bandwidth
- Signal-to-Noise Ratio (SNR)
- Modulation
- Error Rate
- Propagation Delay
- Attenuation
- Multipath Fading
- Channel Capacity
- Duplexing
- Protocol Overhead

Communication Systems Parameters: *Baud Rate vs. Bit Rate*

- **Baud Rate**
  - **Definition:** The **number of signal or symbol changes per second** in a communication channel. It indicates the rate at which information is transferred.
  - **Units:** Measured in baud (symbols per second) or bps (bits per second).



- **Bit Rate (Data Rate)**
  - **Definition:** The **number of bits transmitted per unit of time**. It represents the actual information-carrying capacity of the communication channel.
  - **Units:** Measured in bps (bits per second).



Communication Protocols

Definition

- Communication protocols define the **rules for the exchange of information** between devices or systems.
- They specify how data is **formatted**, **transmitted**, **received**, and **acknowledged**.

- ***In embedded systems,*** communication protocols are essential for devices to exchange information seamlessly.
- Microcontrollers, sensors, actuators, and other embedded components often adhere to specific communication protocols to ensure compatibility and proper functioning within a larger system.

Communication Protocols

Components of Communication Protocols

- **Syntax:** Defines the structure and format of the data. It includes specifications for data encoding, such as how bits and bytes are organized.
- **Semantics:** Specifies the meaning of each data element, ensuring that the transmitter and receiver interpret information in the same way.
- **Timing:** Governs the timing and sequencing of data transmission, addressing issues like when data should be sent and how devices synchronize their communication.

Communication Protocols

Components of Communication Protocols

- **Low-Level Protocols:** Govern the physical and electrical aspects of communication, ensuring that devices can physically transmit and receive signals.
  Examples include RS-232, USB, and Ethernet protocols.
- **High-Level Protocols:** Operate at a higher level, dealing with issues like data integrity, addressing, and routing.
  Examples include TCP/IP (Transmission Control Protocol/Internet Protocol), HTTP (Hypertext Transfer Protocol), and MQTT (Message Queuing Telemetry Transport).

Communication Protocols

Components of Communication Protocols

- Universal Asynchronous Receiver/Transmitter (UART)

  -> Tx/Rx

- Serial Peripheral Interface (SPI)

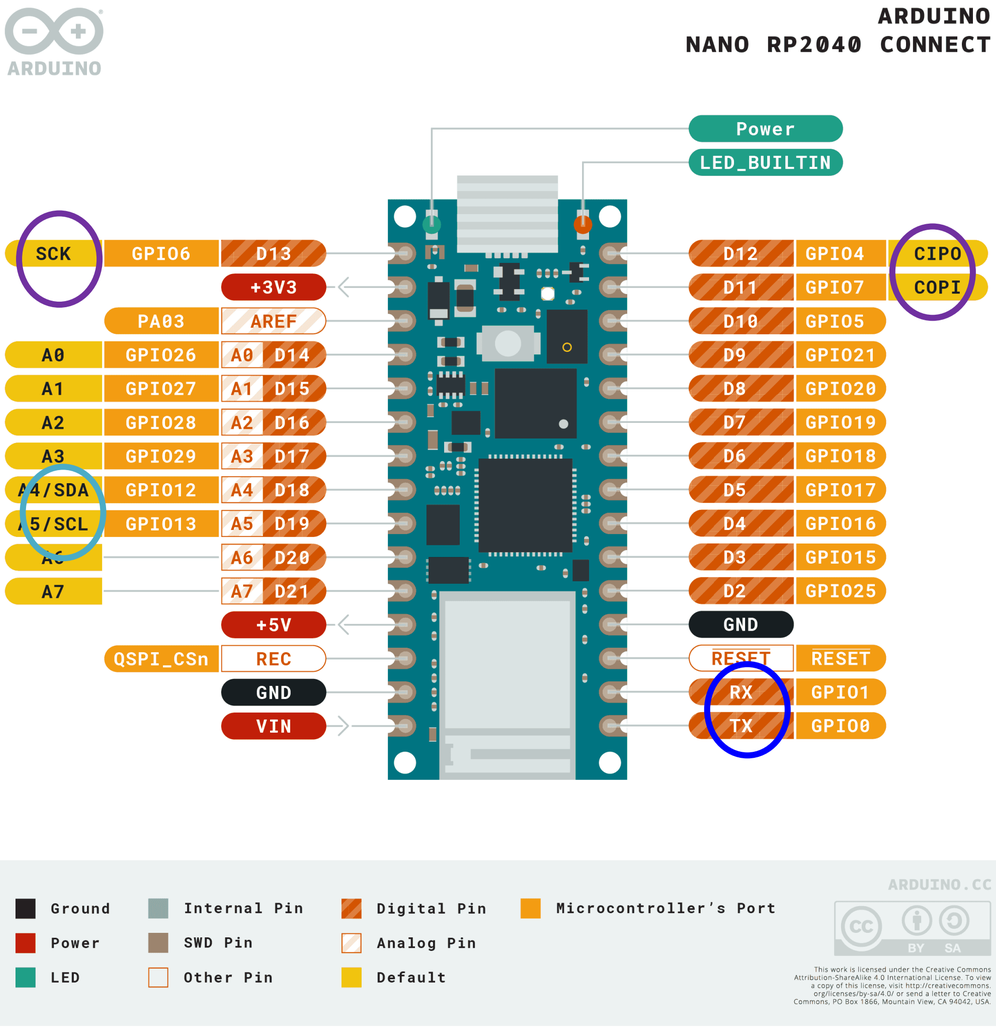
  -> CIPO/COPI/SCK/CS (any GPIO except A6/A7)

- Inter-Integrated Circuit (I2C)

  -> SCL/SDA

- Controller Area Network (CAN)
- Others

Arduino RP2040



Definition

- A **serial** communication protocol commonly used for **point-to-point** communication between devices.
- It is commonly used for Interfacing with sensors, communication between microcontrollers, debugging and configuration interfaces.

Requirements

- Synchronous or Asynchronous?
  - ❑**Asynchronous** which means It uses **two wires** (TX and RX) for serial data transfer
- duplex?
  - ❑Simplex/0.5Dup/Fdup
- Protocol Syntax?

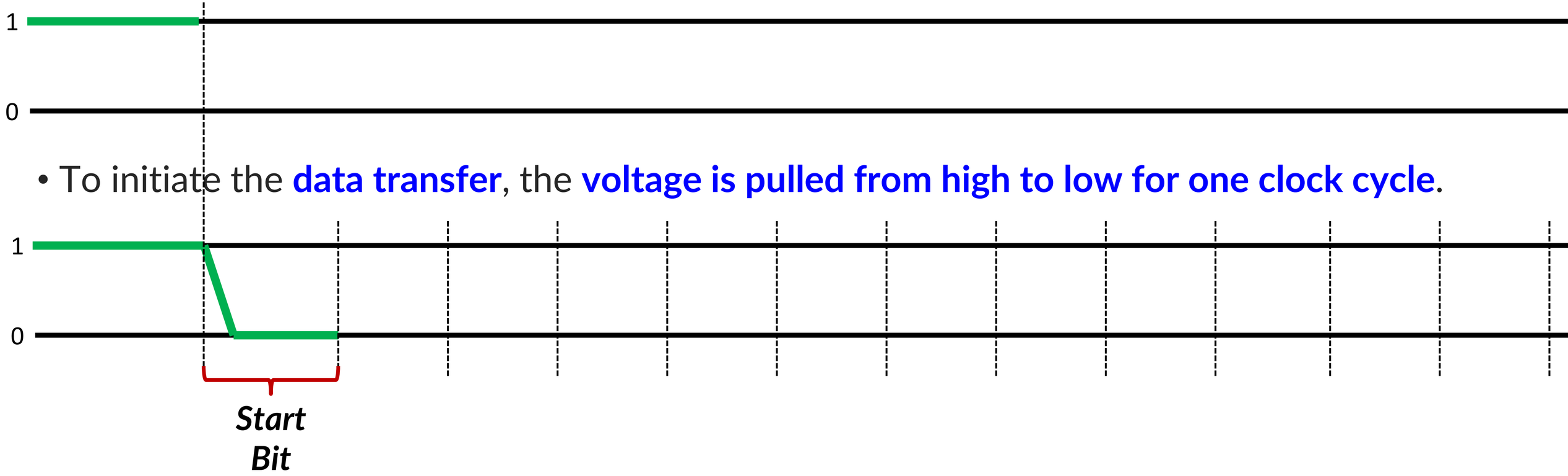
Packet

Start Bit (0)	**Data Bits** (1:5-9)	**Parity Bit** (6-10:7-11)	**Stop Bits** (8-12:9-10/13-14)
1 bit	5-9 bits	0-1 bit	1-2 bits

Data Frame

Start Bit

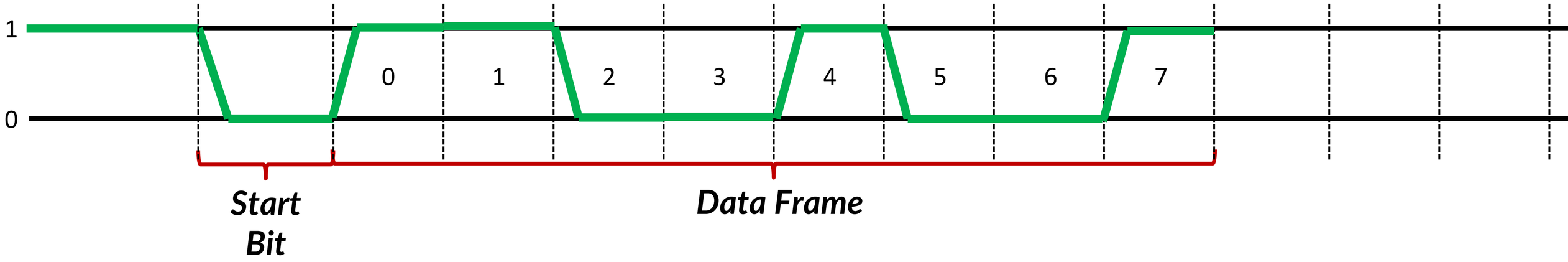
- When there is **no data** transmission, the UART transmission line is held at **high voltage**.



- To initiate the **data transfer**, the **voltage is pulled from high to low for one clock cycle**.
- And when the receiving UART detects the high to low voltage transition, it begins reading the data frame at **the frequency of the baud rate**.

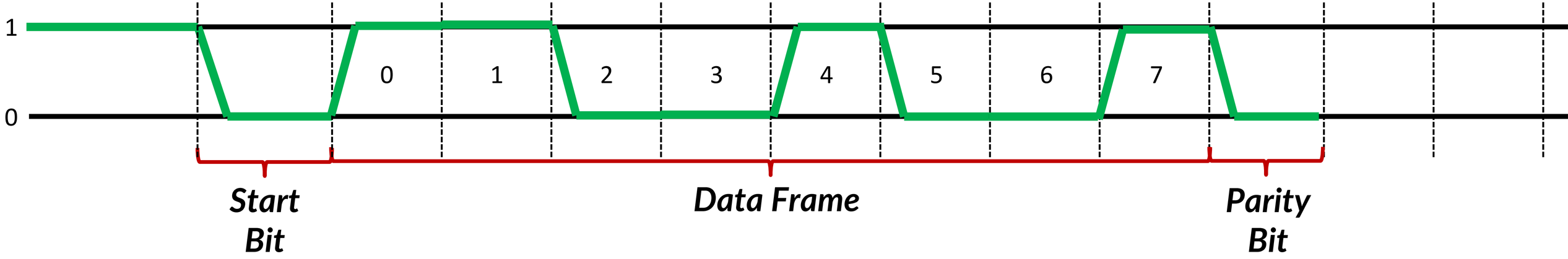
Data Frame

- The data bits are usually **5 to 8** in number.
- If **no parity** bit is used, it can be **9-bit** long.
- In general case, the **least significant bit** of the data is **transmitted first**.
- This is the useful data we're actually transmitting.



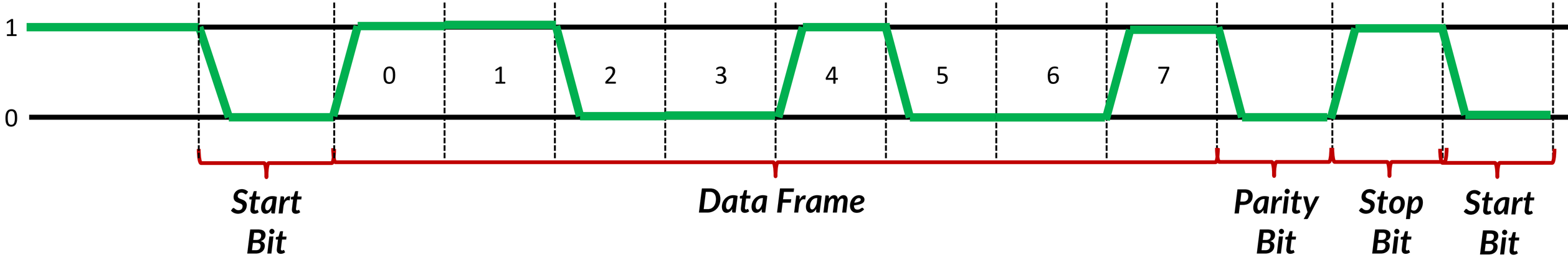
Parity Bit

- A parity bit is used to **indicate potential change in data during transmission**.
- After receiving the data frame, UART checks if the total number of bits with a value of **1** in the data frame is an **even** or **odd** number.
- If the parity bit is
  - **0**, the ones is an **even** number. → $P_{even} = b_7 \oplus b_6 \oplus b_5 \oplus b_4 \oplus b_3 \oplus b_2 \oplus b_1 \oplus b_0 \oplus 0$
  - **1**, the ones is an **odd** number. → $P_{odd} = b_7 \oplus b_6 \oplus b_5 \oplus b_4 \oplus b_3 \oplus b_2 \oplus b_1 \oplus b_0 \oplus 1$



Stop Bit

- To mark the **end** of the data packet, the sending UART drives the data transmission line from a **low voltage to a high voltage.**



Then we can make another start for another byte message

Protocol Syntax Summary

- **Start Bit:**
  - **Purpose:** Marks the beginning of a data frame.
  - **Voltage Level:** Transitions from high to low.
- **Data Bits:**
  - **Purpose:** Represents the actual data being transmitted.
  - **Number of Bits:** Typically 8 bits (can be 5, 6, 7, or 8 bits).
  - **Order:** LSB (Least Significant Bit) or MSB (Most Significant Bit) first.
- **Parity Bit (Optional):**
  - **Purpose:** Provides a basic form of error checking.
  - **Options:** None, Even, or Odd parity.
  - **Position:** Before or after the data bits.
- **Stop Bits:**
  - **Purpose:** Marks the end of a data frame.
  - **Number of Bits:** Typically 1 or 2 bits.
  - **Voltage Level:** Transitions from low to high.

For Further Inquiries, Please send an email

Catherine.elias@guc.edu.eg,
Catherine.elias@ieee.org

Thank you for your attention!

See you next time ☺