Universal Asynchronous Receiver/Transmitter (UART)

Introduction

- The **UART (Universal Asynchronous Receiver/Transmitter)** protocol is a hardware communication protocol used for serial communication between devices.
- It is a widely used standard for low-speed, short-distance communication, such as between microcontrollers, sensors, and other peripherals.
- It works asynchronously, meaning there is no clock signal shared between the devices—each device must know the baud rate (the rate of data transmission) for successful communication.

Key features

- **Asynchronous**: No clock is shared between the sender and receiver. The devices must agree on a baud rate, data bits, stop bits, and parity beforehand.
- **Full-Duplex**: Communication can happen in both directions simultaneously, using separate transmit and receive lines.
- **Data Frame Structure**: The transmitted data is divided into frames that typically consist of:
  - **Start bit**: Marks the beginning of a frame (1 bit).
  - **Data bits**: Usually 5 to 9 bits, depending on the configuration.
  - **Parity bit** (optional): Used for error checking (even or odd parity).
  - **Stop bit(s)**: 1 or 2 bits indicating the end of the frame.
- **Baud Rate**: The rate at which data is transmitted, measured in bits per second (bps). Both devices must be set to the same baud rate for reliable communication.
- **Signal Lines**: Typically, UART uses two lines:
  - **TX (Transmit)**: Sends data from the transmitter to the receiver.
  - **RX (Receive)**: Receives data from the transmitter.

UART functions

- Outbound data
    - Convert from parallel to serial
    - Add start and stop delineators (bits)
    - Add parity bit
- Inbound data
    - Convert from serial to parallel
    - Remove start and stop delineators (bits)
    - Check and remove parity bit

Common Applications

- **Microcontrollers**: For communication with peripherals (e.g., GPS, sensors, Bluetooth modules).
- **Serial ports**: Often used in computers for legacy communication with modems or other serial devices.
- **Embedded systems**: UART is common in embedded systems for debug outputs or communicating with external modules.

Advantages

- Simple, low-cost implementation.
- Wide support in many microcontrollers and devices.
- No need for synchronization between devices beyond the agreed-upon baud rate.

Limitations

- Limited to short-distance communication (usually a few meters).
- The data transfer rate is relatively low compared to other protocols like SPI or I2C.