ASSIGNMENT 4

UART, I²C, SPI, CAN, and USB are common communication protocols in embedded systems, each designed with unique features that make them suitable for specific applications. Here’s a breakdown of their differences based on speed, complexity, pin usage, and application:

**1. UART (Universal Asynchronous Receiver/Transmitter)**

* **Data Transmission Speed**: Generally up to 1 Mbps, but speeds vary depending on the application and hardware.
* **Complexity**: Simple and easy to implement since it requires minimal configuration.
* **Pin Usage**: Uses only two pins (TX for transmit, RX for receive), but each device requires its own set of UART pins.
* **Device-to-Device Communication**: Primarily point-to-point, meaning it supports one-to-one communication (half-duplex or full-duplex).

**Key Features and Applications**

* **Best suited for** low-speed communication where simplicity is valued.
* **Common in** serial communication with peripherals like GPS modules, Bluetooth modules, and sensors in microcontrollers.
* **Use Case Example**: Communication between a microcontroller and a terminal console or between a GPS module and an embedded device.

**2. I²C (Inter-Integrated Circuit)**

* **Data Transmission Speed**: Typical speeds are 100 kbps to 400 kbps; some fast modes go up to 3.4 Mbps.
* **Complexity**: Moderate complexity; requires addressing each device on the bus.
* **Pin Usage**: Uses two pins (SDA for data, SCL for clock), allowing multiple devices to share the same bus.
* **Device-to-Device Communication**: Supports multiple devices on a single bus, with master-slave communication.

**Key Features and Applications**

* **Ideal for** connecting multiple peripherals like sensors, EEPROMs, RTCs, and displays in a master-slave configuration.
* **Common in** sensor networks or applications where multiple peripherals need to communicate with a single microcontroller.
* **Use Case Example**: Temperature sensors, EEPROM storage, and real-time clocks in a microcontroller-based application.

**3. SPI (Serial Peripheral Interface)**

* **Data Transmission Speed**: Up to 10 Mbps or more, depending on the device and system requirements.
* **Complexity**: Moderate; each device needs a separate chip select (CS) line.
* **Pin Usage**: Minimum of four pins (MOSI, MISO, SCK, and CS), with additional CS pins for each device connected.
* **Device-to-Device Communication**: Supports master-slave communication, typically one master to multiple slaves with a separate CS line for each slave.

**Key Features and Applications**

* **Best for** applications requiring high-speed, full-duplex communication over short distances.
* **Common in** applications where speed is critical, such as SD card data transfer, high-speed sensors, and TFT displays.
* **Use Case Example**: Connecting a microcontroller to an LCD display, SD card, or fast data-acquisition sensors.

**4. CAN (Controller Area Network)**

* **Data Transmission Speed**: Typically 1 Mbps (classic CAN); CAN FD (Flexible Data-Rate) allows for faster data rates in the data phase.
* **Complexity**: High; requires specialized CAN transceivers and controllers.
* **Pin Usage**: Typically uses two wires (CAN\_H and CAN\_L) with differential signaling.
* **Device-to-Device Communication**: Multi-master protocol with robust error-checking, supporting multiple devices on the same bus.

**Key Features and Applications**

* **Designed for** reliability and error-checking in noisy environments, especially for automotive and industrial applications.
* **Common in** automotive systems (e.g., connecting ECUs, sensors, and actuators), as well as industrial automation.
* **Use Case Example**: In-car communication between engine control, braking systems, and transmission.

**5. USB (Universal Serial Bus)**

* **Data Transmission Speed**: Can range from 1.5 Mbps (Low Speed) to 10 Gbps (USB 3.1 Gen 2).
* **Complexity**: High; requires a dedicated USB controller with complex handling of enumeration and protocol layers.
* **Pin Usage**: Minimum of four pins (VCC, GND, D+, D-); some versions have additional pins for faster speeds.
* **Device-to-Device Communication**: Host-device communication model, usually one-to-one but can support multiple devices with a hub.

**Key Features and Applications**

* **Used for** high-speed data transfer, often for user interface devices, storage, and general peripheral connectivity.
* **Common in** personal computers and embedded systems that need USB connectivity, such as flash drives, cameras, or mouse/keyboard interfaces.
* **Use Case Example**: Data transfer between a microcontroller-based system and a PC or other USB devices.