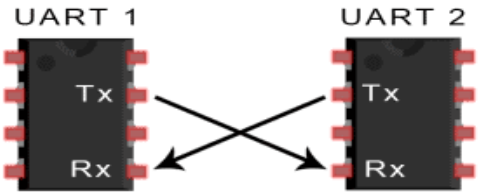
Universal Asynchronous Receiver and Transmitter (UART) Protocol

UART (Universal Asynchronous Receiver/Transmitter) is a hardware communication protocol used for serial data transmission between devices. It transfers data bit by bit (serially) without needing a shared clock signal, which makes it asynchronous.

Signals:

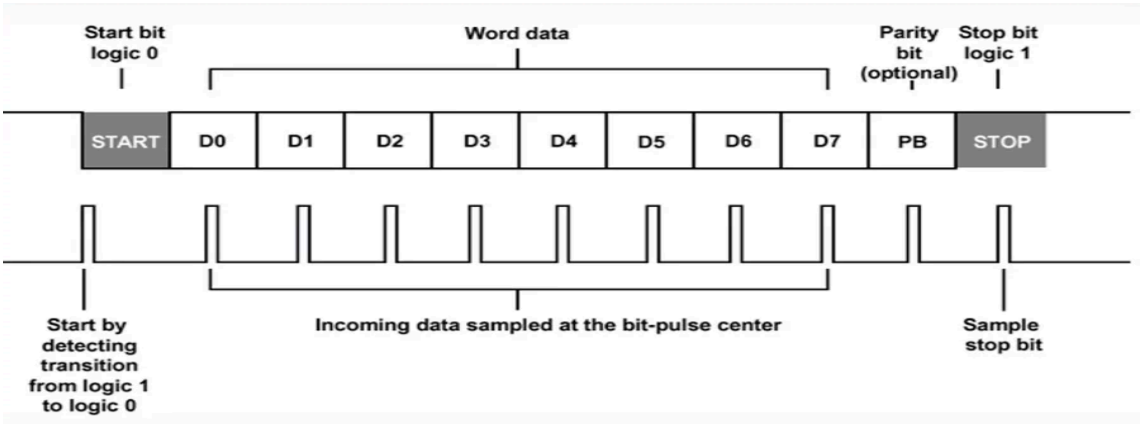
- TX (Transmit Data): Serial output line that sends bits.
- RX (Receive Data): Serial input line that receives bits from TX of the other device.



Features:

- Data Order → Always LSB first (not configurable).
- Full Duplex →TX and RX lines work simultaneously.
- Frame Size → Commonly 8 bits, but can be 5 to 9 bits depending on configuration.
- Speed → Typically 300 bps to 1 Mbps, some controllers support up to ~5–10 Mbps.
- Parity → Optional bit for error detection (Even, Odd, None).
- No ACK/NACK → Unlike I²C, UART has no acknowledgment mechanism.

Frame Format:



Baud Rate:

- The rate at which symbols (bits) are transmitted over UART.
- Baud ticks are not a clock signal; they are just pulses derived from the system clock.
- Example: At 115200 baud, you get 115200 baud ticks per second.

Formula for Baud Divisor:

$$Divisor = \frac{f_{clk}}{Baud\_Rate}$$

Application:

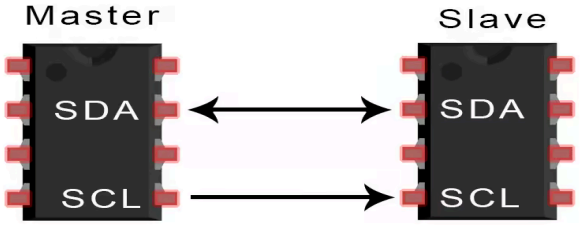
- Debugging Interfaces – Serial console for microcontrollers, FPGAs, and Linux boards.
- Communication with Modules – GPS, GSM, Wi-Fi, Bluetooth, RFID, etc.
- Programming/Flashing Devices – Used in bootloaders to upload firmware.
- PC to Embedded System – Via USB-to-UART converters (FT232, CP2102, CH340, etc.).

Inter-Integrated Circuit (I2C) Protocol

The I2C (Inter-Integrated Circuit) protocol is a master-slave, synchronous serial communication protocol for short-distance, chip-to-chip communication between microcontrollers and peripherals

Signals

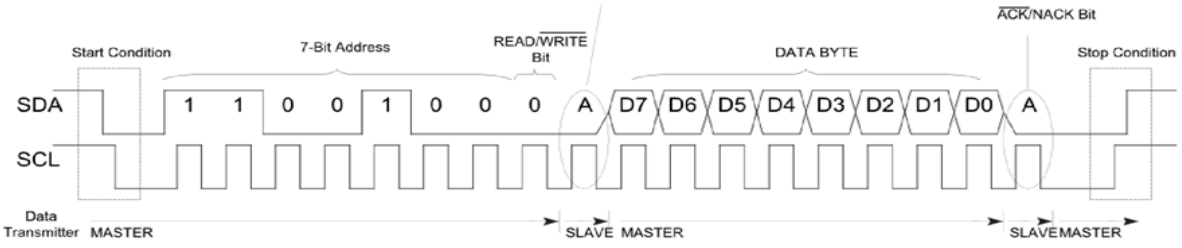
- SCL – Serial clock is used to synchronise the data flow.
- SDA – Serial data is the data line that can send and receive data.
- Both the pins are bidirectional and open drain. Pull-up resistors keep the line HIGH.An open-drain output is a pin that can only pull the line LOW (to ground).



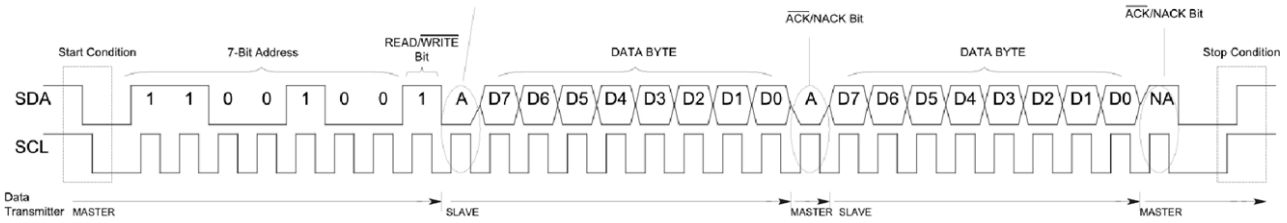
Features:

- Data Order → Always **MSB** first (not configurable).
- Half-Duplex → One bidirectional data line (SDA) is shared for transmit/receive.
- Frame Size → Fixed **8-bit data** frames, followed by ACK/NACK.
- Speed → Standard (100 kbps), Fast (400 kbps), Fast+ (1 Mbps), High-Speed (3.4 Mbps).
- ACK/NACK → Every byte is acknowledged (ACK) or not (NACK) by the receiver.
- Addressing → 7-bit (standard) or 10-bit (extended) slave addresses.

Frame format:



Write transaction



Read transaction

Data Transmission:

- Idle State → Both SCL and SDA are HIGH.
- Start Condition (S) → SDA goes LOW while SCL is HIGH.
- Address Phase → Master sends slave address (7 or 10 bit) + R/W bit.
- ACK/NACK → Slave responds with ACK (0) or NACK (1).
- Data Phase → 8-bit data sent MSB first, followed by ACK/NACK.
- Repeated Start (Sr) → Master issues another START without STOP, especially for read transactions.
- Stop Condition (P) → SDA goes HIGH while SCL is HIGH.

Application:

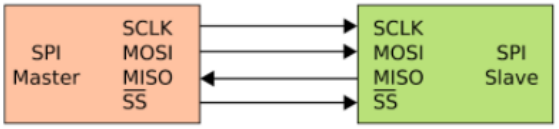
- Multi-master, multi-slave → Multiple masters and slaves on the same bus.
- Clock stretching → Slaves can slow down the master if needed.
- EEPROMs, Flash memory ,RTC (Real-Time Clock) , Temperature and humidity sensors

Serial Peripheral Interface (SPI) Protocol

SPI (Serial Peripheral Interface) is a very simple, high-speed, synchronous serial protocol

Signals

- SCLK – Serial Clock (from master)
- MOSI – Master Out, Slave In
- MISO – Master In, Slave Out
- CS/SS – Chip Select (active low, selects slave)



Features:

- Data Order → Default is MSB first, but configurable as LSB first.
- Full-Duplex → Data out (MOSI) and in (MISO) occur simultaneously.
- Frame Size → Commonly 8 bits, but can also be 4, 16, 32, etc.,
- Speed → Typically 10–50 Mbps, some controllers support 100+ Mbps.
- No ACK/NACK → Unlike I²C, SPI has no acknowledgment.
- Chip Select (CS/SS) → Defines transaction boundaries (active low).

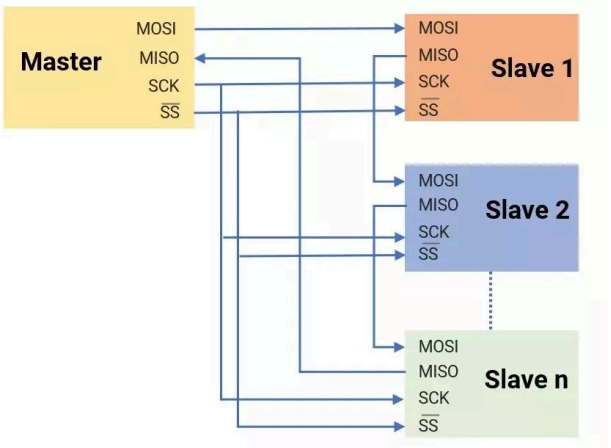
Key Configurations

Clock Polarity (CPOL)
CPOL=0: Clock idle = Low
CPOL=1: Clock idle = High

Clock Phase (CPHA)
CPHA=0: Data sampled on 1st clock edge
CPHA=1: Data sampled on 2nd clock edge

For example: In Mode 3:
CPOL = 1, which means the clock is idle high.
CPHA = 1, which means the data is sampled on the second clock edge.
So, the first edge after the idle high state is a falling edge, where the data changes.
The data is then sampled on the rising edge of the clock.

Daisy-chain configuration:
All slaves are connected in series, with the MISO of one slave linked to the MOSI of the next.
A single CS and SCK control all devices, and data shifts through the chain so each slave captures its portion while passing the rest along. This reduces chip select lines but requires careful bit-count management.



Application: Sensors like Gyroscopes, accelerometers, temperature sensors, etc. ADC/DAC chips, EEPROMS and TFT display.

Comparison of UART, I²C, and SPI Protocols

Feature / Protocol	UART	I²C	SPI
Company	Originally by Texas Instruments	Philips (now NXP)	Motorola (now NXP and others)
Type	Asynchronous serial	Synchronous serial	Synchronous serial
Wires / Signals	2 → TX, RX	2 → SDA, SCL	4 → SCLK, MOSI, MISO, CS/SS
Clock	No shared clock (baud rate)	Master provides SCL	Master provides SCLK
Direction	Full-duplex (TX & RX independent)	Half-duplex (SDA shared)	Full-duplex (MOSI & MISO simultaneously)
Data Order	Always LSB first	Always MSB first	Default MSB first, configurable LSB first
Frame Size	5–9 bits configurable	8-bit frames	Typically 8 bits, can be 4, 16, 32, etc.
Frame Format	Start + Data + Parity (optional) + Stop	Start + Address (7/10 bit) + R/W + Data + Stop	Start + Data (no standard parity/stop, controlled by CS/SS)
Masters / Slaves	One-to-one	Multi-master, multi-slave	Single master, multiple slaves
Line Type	Push-pull	Open-drain with pull-ups	Push-pull
Relative Speed	Slow	Medium	Fast
Applications	Debugging, serial console, GPS/GSM/Wi-Fi modules, PC ↔ MCU	Sensors, EEPROM, RTC, LCDs, GPIO expanders	High-speed sensors, ADC/DAC, Flash memory, displays, TFTs