SJSU SAN JOSÉ STATE UNIVERSITY

Charles W. Davidson College of Engineering

Department of Computer Engineering

Real-Time Embedded System Co-Design
CMPE 146 Section 1
Fall 2024

MCU Digital Interfaces

Serial Communication

- Sending digital data one bit at a time over a communication channel
  - Contrast to parallel communication: multiple bits are sent in parallel at the same time over multiple channels
- Traditionally used for long-distance (> tens of feet) communication
  - Fewer conductors compared to parallel communication
    - Cable cost is much less
    - Cable occupies less space
  - No synchronization issue among multiple data channels
  - Crosstalk issue is not as severe due to fewer conductors
- Has become choice of communication even for short distances
  - Demand for increasing higher bandwidth makes parallel communication not practical
    - Keeping parallel channels aligned requires careful channel implementation
    - Crosstalk issue becomes more severe
  - Improvements in electronics make implementation more practical
    - Serialization and deserialization overheads much reduced
    - Much higher frequency can be used
  - Improvements in the physical materials allows higher frequencies to be used

Serial Communication (cont'd)

- Well suited for embedded applications
  - On-board chip-to-chip communication as well as off-board situations
  - Space is much limited
    - MCUs have very limited number of pins
    - Most systems are small; connectors cannot be big
  - Very high bandwidth is not needed for many applications
  - Less cost in components and cables
- Various physical media can be used for the actual transmission
  - Different characteristics in bandwidth, delay, cost, and ease of installation and maintenance
  - Guided media
    - Copper wires, optical fibers, etc.
  - Unguided media
    - Air, vacuum, water, etc.

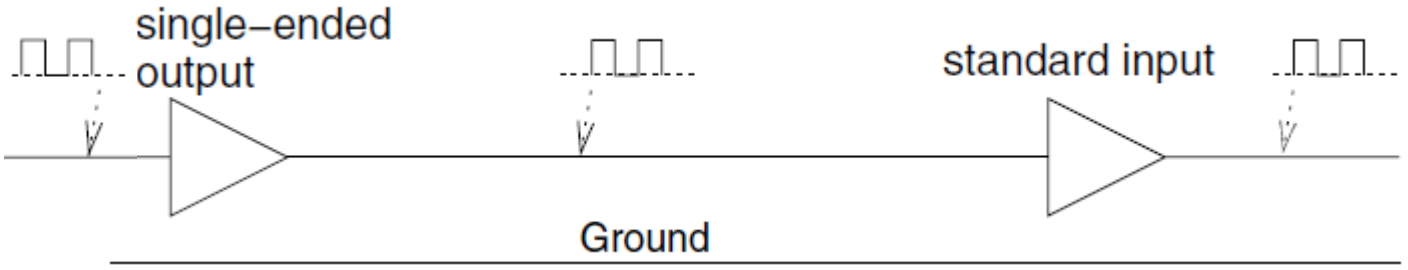
Serial Communication Basics

We focus on digital communications using wire conductors

- Transmission means sending digital data as a series of voltage levels down a wire to a receiver
- Asynchronous vs. synchronous
  - Synchronous: both sender and receiver use the same clock signal when send to and read from the data line
  - Asynchronous: Reading of the data line is initiated by detecting a signal pattern
    - No sharing of clock signal
- Modes of transmission
  - Simplex: Data are transferred from transmitter to receiver and not vice versa
  - Half Duplex: Transmission can occur in only one direction at a time
  - Full Duplex: Transmission and reception of data can occur at the same time
    - Requires separate and independent physical data channels
- Baud Rate
  - Number of voltage level changes per unit time
    - Each level can represent multiple bits
  - For communication schemes under discussion, bit rate = baud rate

Signaling Methods

- Single-ended signaling
  - There is a reference signal (typically ground) to all signals carried on the lines
    - Both transmitting and receiving devices share the same reference
  - Noise can be picked up along the signal path
    - Using higher Logic-1 voltage or lower Logic-0 can improve noise immunity



- Differential signaling
  - There is no reference signal as in single-ended signaling
  - Each signal is carried by two wires: one is the "negative" of the other

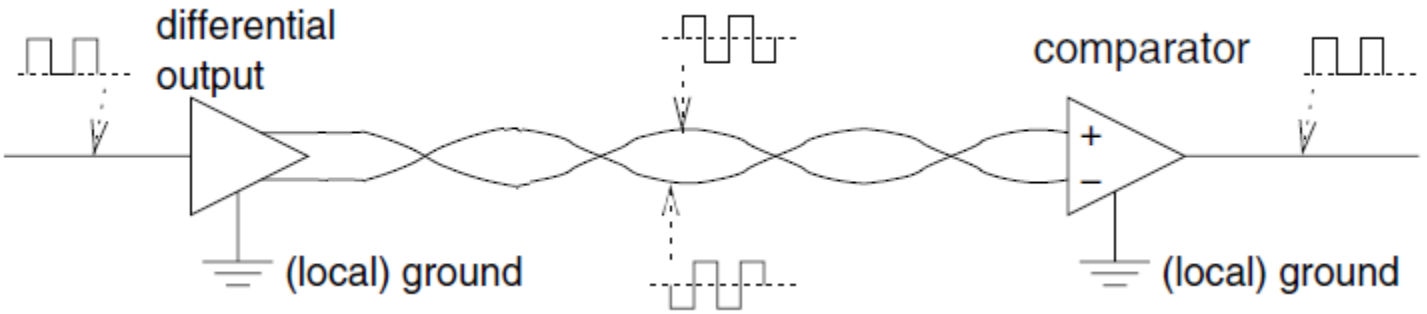
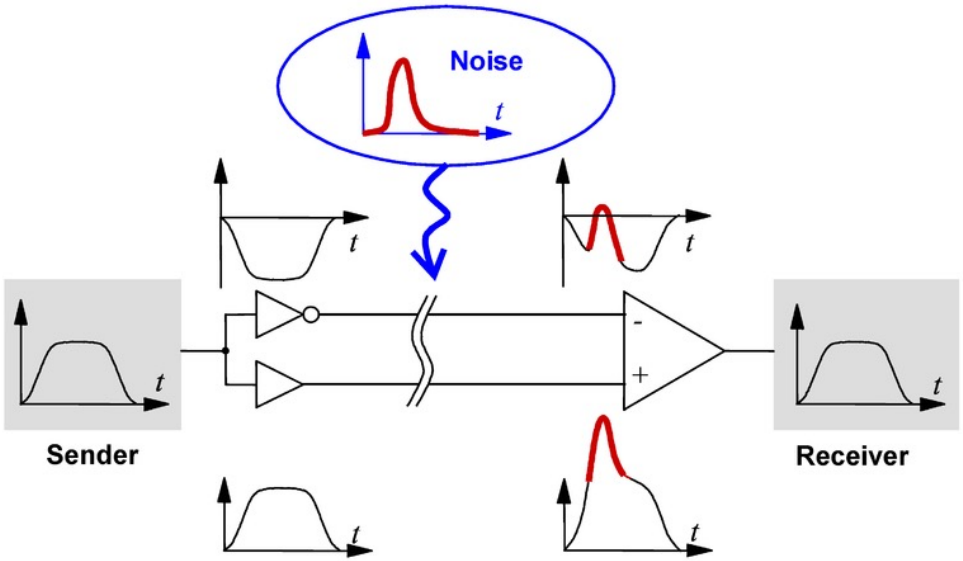
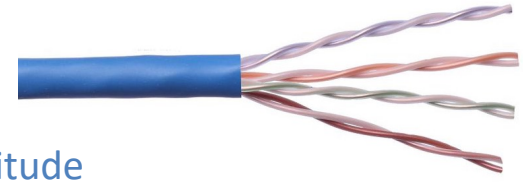


Diagram source: Embedded System Design, Marwedel, Springer, 2006

Signaling Methods (cont'd)

- Differential signaling
  - Inherent ability to reject noise picked up along the signal path



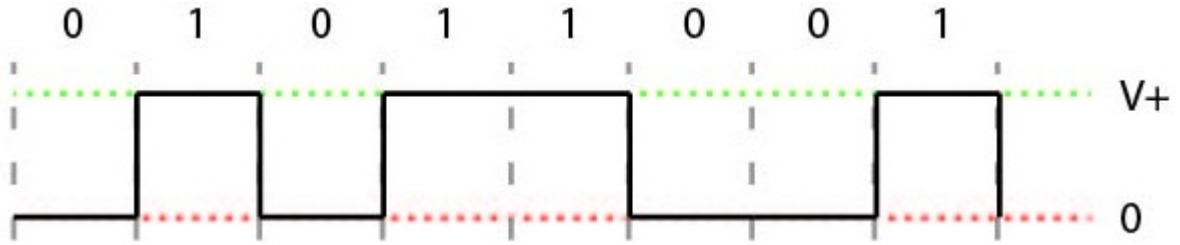


  - Wires in the pair are twisted
    - Induced voltages on both lines would be close in magnitude
  - Drawbacks
    - Higher cost: Number of lines doubled
    - More circuitry

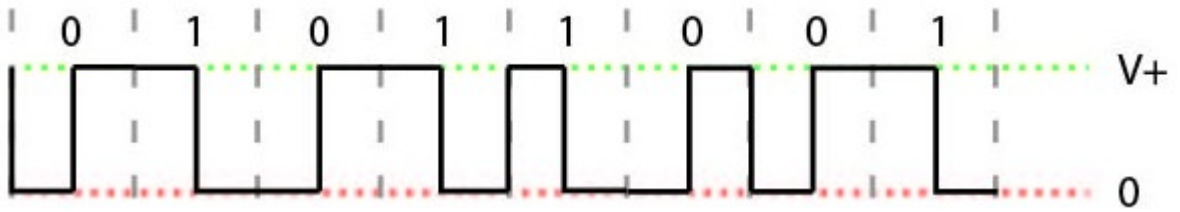
Logic Value Encoding

Two common methods to encode a logic value on a signal line

- Non-return-to-zero Encoding (NRZ non-polar)
  - Logic 1: High voltage level
  - Logic 0: Low voltage level



- Manchester Encoding
  - Logic 1: Falling voltage transition
  - Logic 0: Rising voltage transition

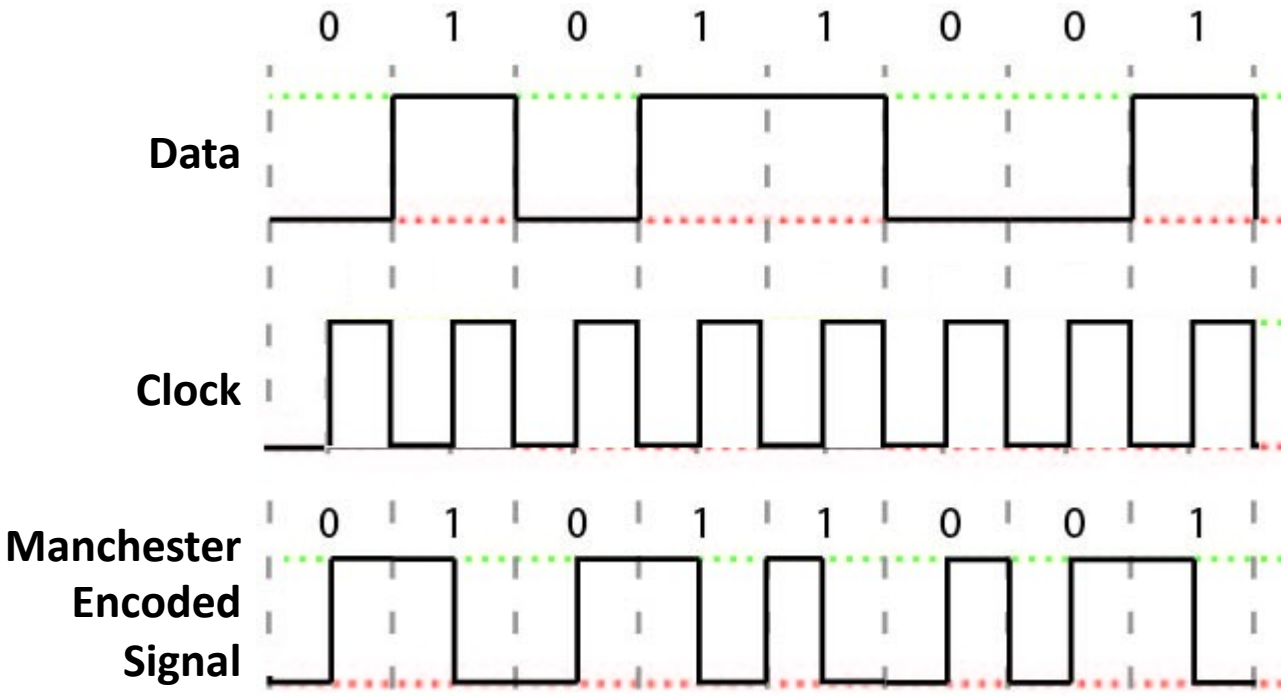


  - A self-clocking scheme: Do not need a separate clock signal to locate data
    - A transition within a clock period indicates valid data

Diagram source: https://www.electronicdesign.com/communications/what-s-difference-between-nrz-nrzi-and-manchester-encoding

Logic Value Encoding (cont'd)

- Manchester Encoding
  - Easy to encode data: Data XOR clock



  - Drawback: Need twice of the bandwidth of NRZ

Diagram source: https://www.electronicdesign.com/communications/what-s-difference-between-nrz-nrzi-and-manchester-encoding

Frame

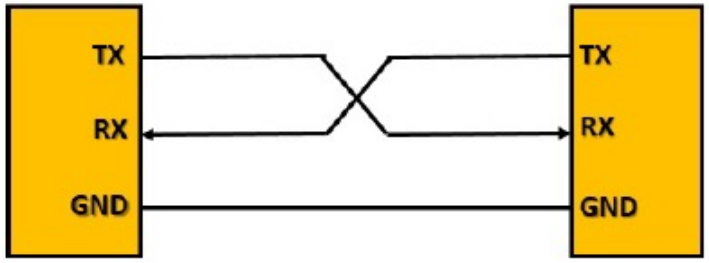
For asynchronous communication

- Data are transmitted in a well-defined frame over the media (wires)
- A frame is a stream of contiguous bits that has three parts

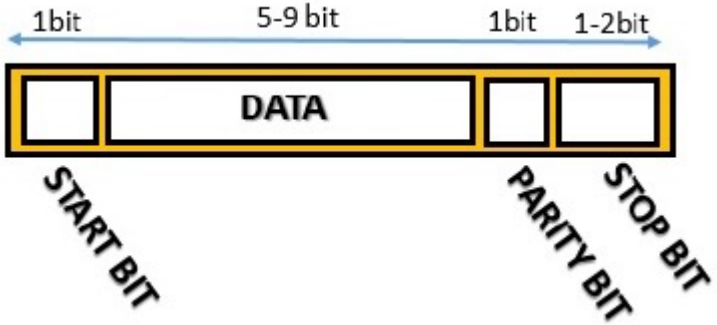
  - Header
    - Certain bit pattern to indicate the beginning of a frame
    - May contain some additional info, e.g., destination address, data length
  - Payload (data)
    - The actual information to be transferred from one device to another
  - Trailer
    - Typically contains checksum to ensure data integrity
    - To create a gap to the next frame
- Receiver constantly looks for the header to identify the presence of a frame on the media

UART

- UART (Universal Asynchronous Receiver/Transmitter)
- Asynchronous, full-duplex, NRZ
- Most basic configuration
  - Three wires: Transmit, receive and ground



- Frame
  - Start bit, data, parity bit, stop bit



- Two common encoding methods
  - TTL (Transistor-Transistor Logic) : Logic 1: +5 V, Logic 0: 0 V
  - RS-232: Logic 1: −15 V to −3 V, Logic 0: +3 V to +15 V

USB

- Universal Serial Bus (USB)
  - We will focus our discussion on USB 2.0
- Widely used for connecting peripherals devices to computers
- 1.5, 12 or 480 Mbps
- Asynchronous, half-duplex
- Four wires
  - Power: V_{BUS} (5 V)
    - Capable of delivering up to 500-mA to the peripheral
  - Ground
  - Two differential signaling lines: D+ and D-
- NRZI (Non-Return-to-Zero Inverted) encoding
  - Logic 1: Transition occurs
  - Logic 0: No transition

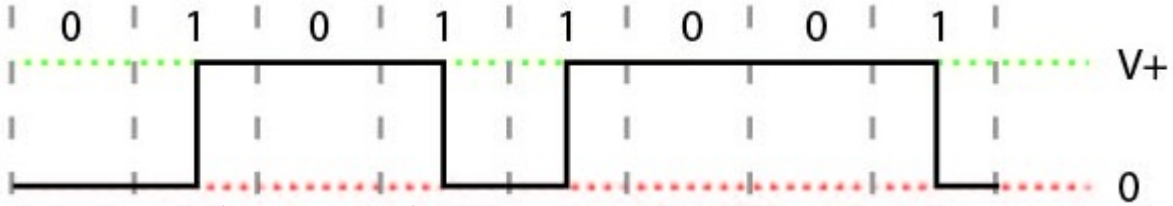
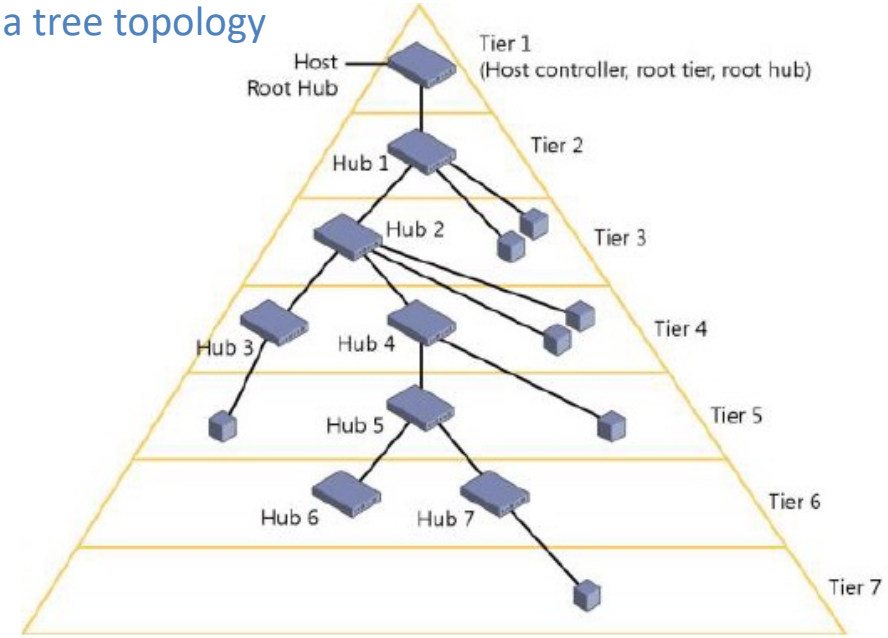


Diagram source: https://www.electronicdesign.com/communications/what-s-difference-between-nrz-nrzi-and-manchester-encoding

USB - Architecture

- Single-Master-Multiple-Slave architecture
  - Link is point-to-point
  - A hub is needed for configuration with more than one slave
    - Bus forms a tree topology



- Master periodically polls slaves to check if there is a need to transmit or receive data
  - Slaves cannot initiate any transactions

Diagram source: Cypress Semiconductor Application Notes: USB 101: An Introduction to Universal Serial Bus 2.0

USB - Transceiver

- Transceiver
  - D+ and D- can be driven and sensed independently

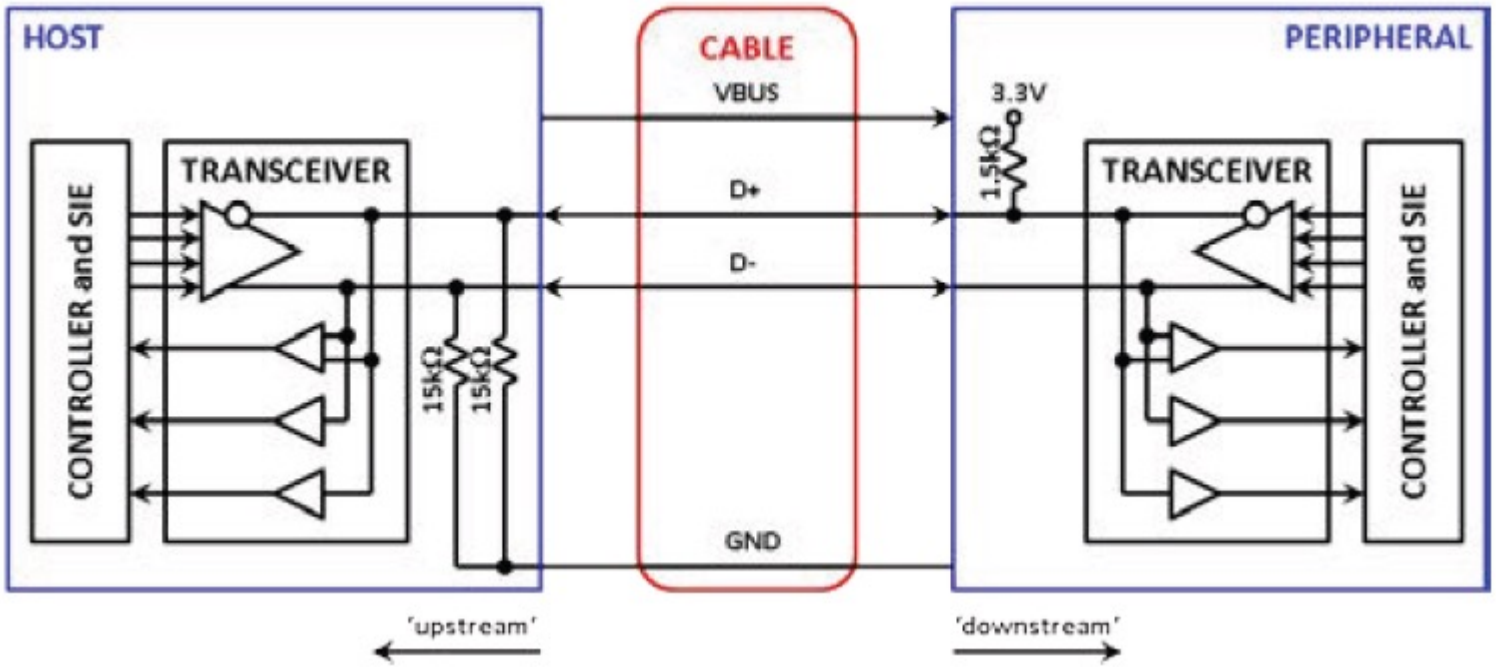
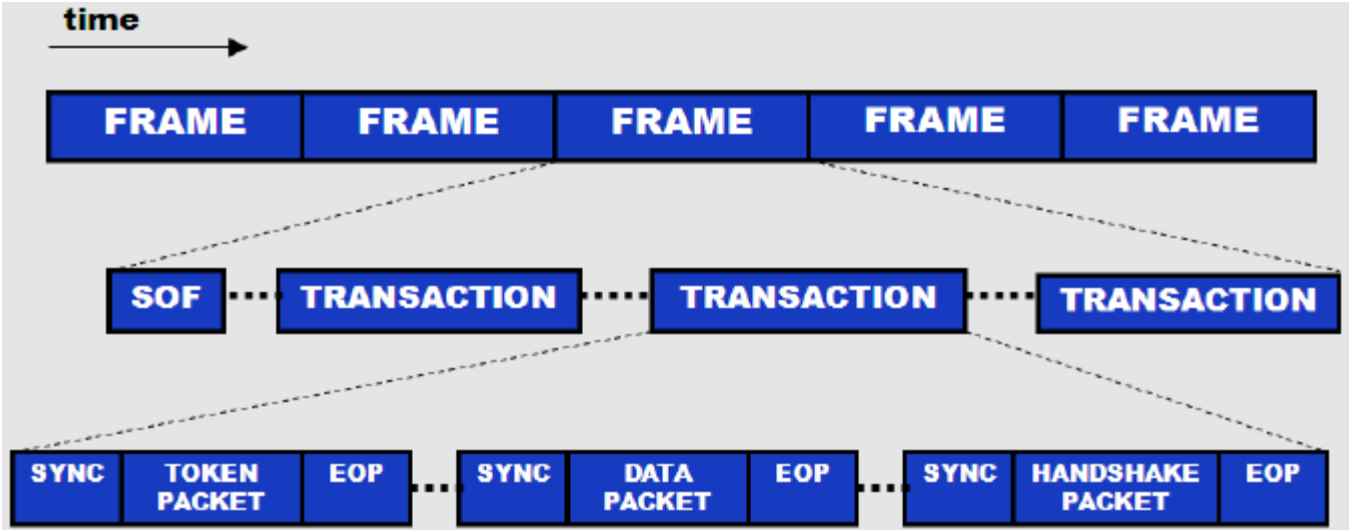


Diagram source: https://www.analog.com/media/cn/technical-documentation/tech-articles/ms-2579_cn.pdf

USB - Frame



- Frame structure
  - Start of Frame (SOF)
  - One or more transactions
    - Each transaction is made up of a series of packets
- Transaction
  - Exchange of packets between master and slave
  - Must start with a token packet followed optionally by
  - Data and/or handshake packets

Diagram source: Cypress Semiconductor Application Notes: USB 101: An Introduction to Universal Serial Bus 2.0

USB - Packet

- Packet structure
  - Preceded with a synchronization (SYNC) pattern (part of header)
    - A bit pattern for receiver to synchronize with transmitter
  - Ends with an End-of-Packet (EOP) pattern (part of trailer)
    - A bit pattern to indicate the end of packet

- Packet format

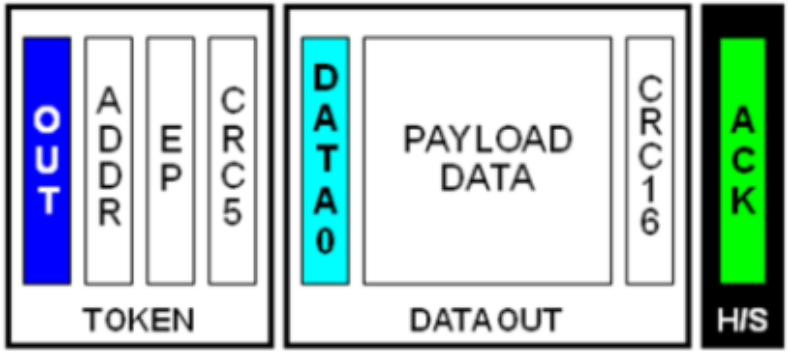
  - PID: Packet ID, 8 bits
    - Identifies the type of packet being sent
  - ADDR: Address, 7 bits, optional
    - Packet destination device address
  - EP: End point address, 4 bits, optional
    - To address some portion within the device
  - PAYLOAD: Data, up to 1024 bytes, optional
  - CRC: checksum, 5 or 16 bits (for data packets), optional

USB – Packet (cont'd)

- Packet types
  - Token packets
    - Initiate transaction, e.g., read, write, setup
    - Identify device involved in transaction
    - Always sourced by the master
  - Data packets
    - Delivers payload data
    - Sourced by master or slave
  - Handshake packets
    - Reply the transaction status or report device status
    - Sourced by receiver of data
  - Special packets
    - Facilitates bus control
    - Sourced by master

USB – Transaction Examples

- Example of write transaction
  - Three packets: Token, data and handshake



- Example of read transaction
  - Three packets: Token, data and handshake

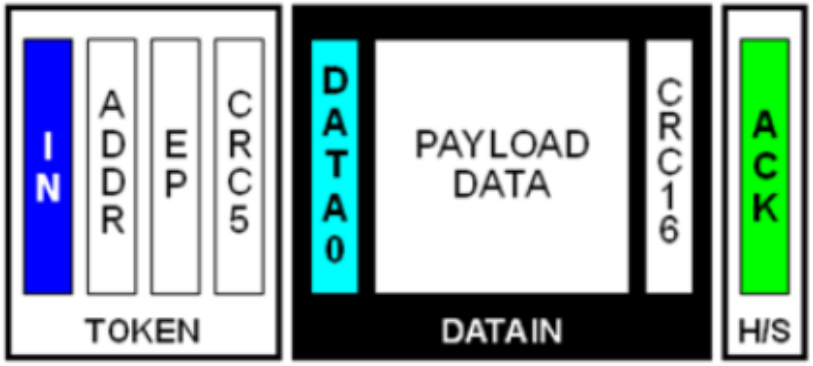


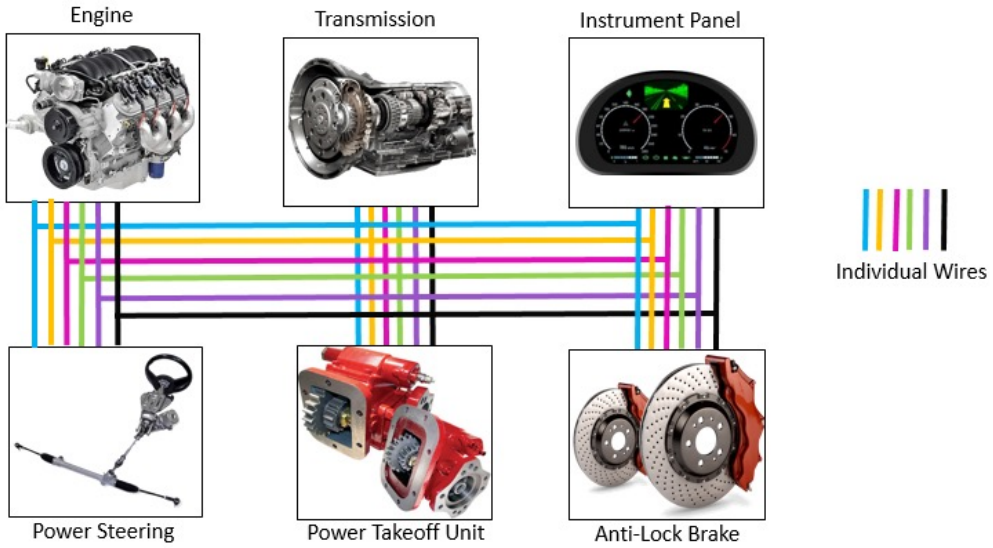
Diagram source: Cypress Semiconductor Application Notes: USB 101: An Introduction to Universal Serial Bus 2.0

CAN

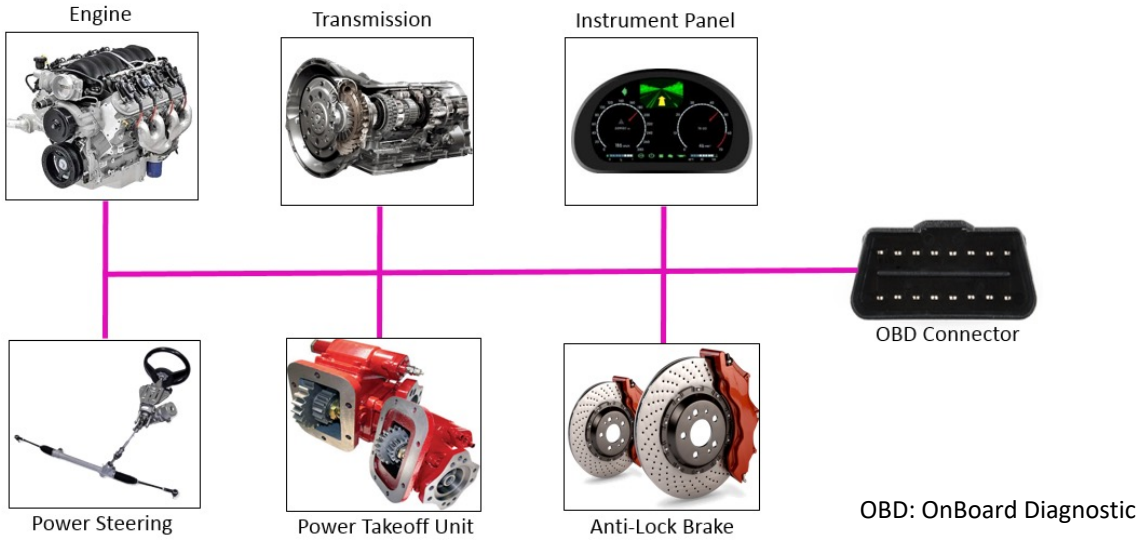
- Controller Area Network (CAN)
- Asynchronous, half-duplex, NRZ
- Maximum data rate: 1 Mbit/s
- Typically used for connecting various standalone modules in large embedded systems
  - Widely used in industrial automation, vehicles, aircrafts, etc.
- Designed for distributed real-time control with high level of reliability
  - Data frames are prioritized for timely delivery of control messages
  - Differential signaling is adopted for handling noisy environment
- Original motivation was to save wires in automobiles
  - Facilitates communication between electronic control units (ECUs), sensors and actuators
  - ECUs communicate with each other without dedicated wiring in between

CAN - Application

- Without CAN



- With CAN



OBD: OnBoard Diagnostic

Diagram source: https://community.sw.siemens.com/s/article/can-bus-measurements

CAN - Architecture

- Two-wire bus
  - Wires: CAN_H and CAN_L
- Multi-master-multi-slave architecture
  - All messages are broadcast; all nodes receive the same messages

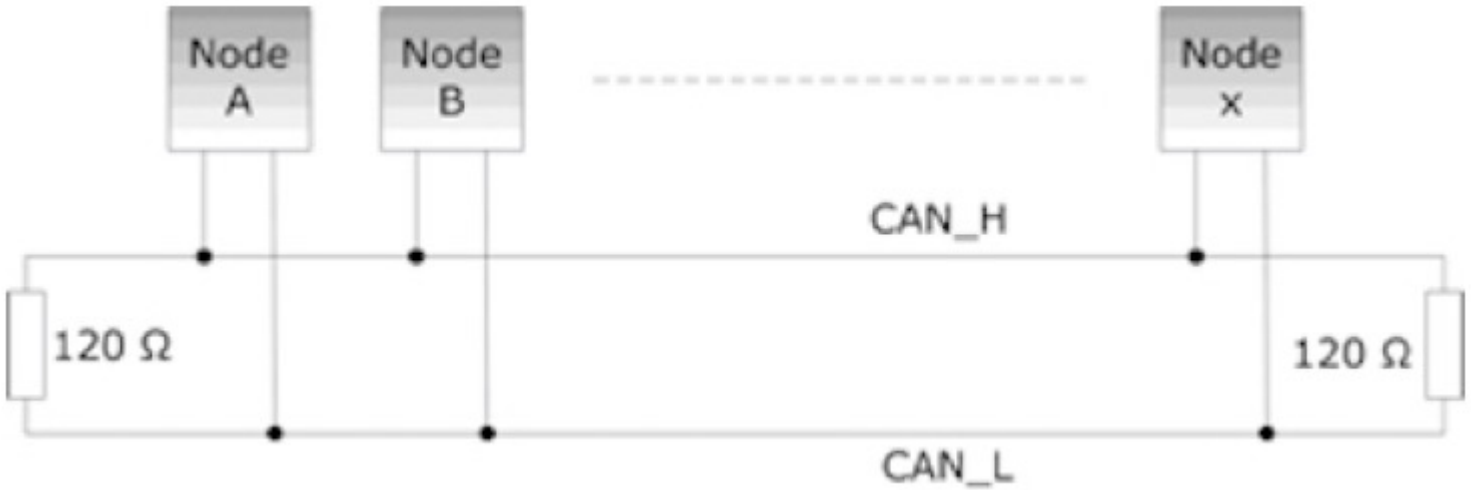
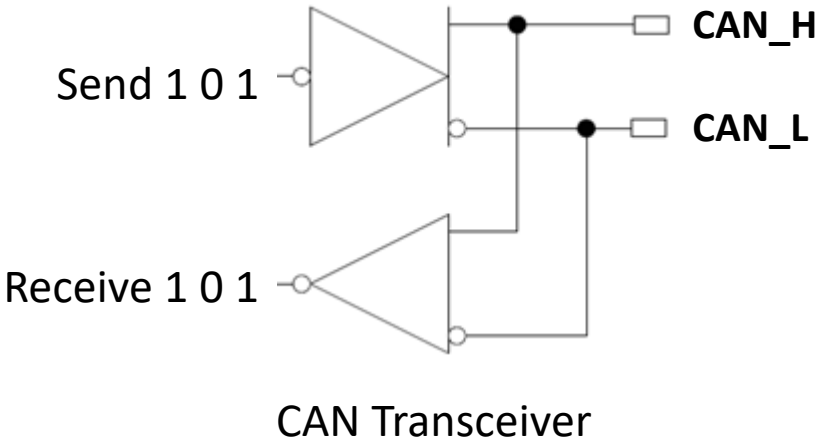


Diagram source: https://copperhilltech.com/a-brief-introduction-to-controller-area-network/

CAN – Signaling Method

- Differential signaling
- Two logic states
  - Recessive (Logic 1): $V_{CAN\_H} \approx V_{CAN\_L} \approx 2.5$ V; $\Delta V \approx 0$ V
  - Dominant (Logic 0): $V_{CAN\_H} \approx 3.5$ V, $V_{CAN\_L} \approx 1.5$ V; $\Delta V \approx 2$ V
  - If a node outputs a 0, the state of the bus changes to 0
    - That is, a dominant bit always overwrites a recessive bit on the bus
    - Essentially an AND operation on all individual output states



CAN Transceiver

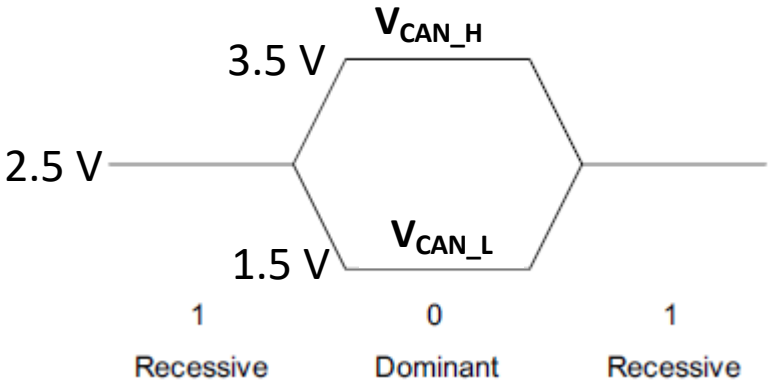
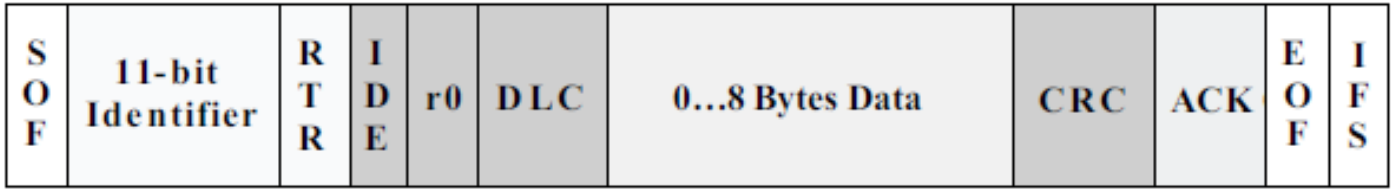


Diagram source: TI Application Report: Introduction to the Controller Area Network (CAN)

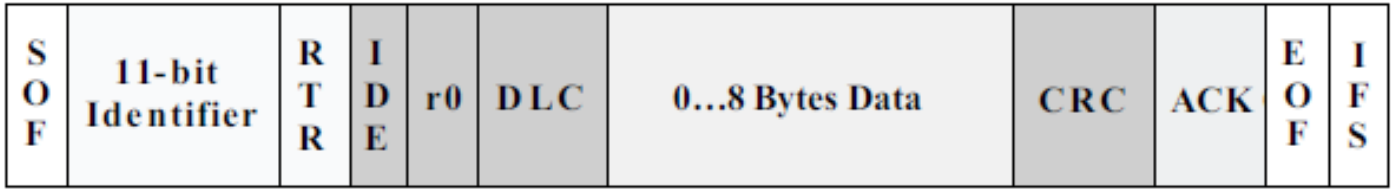
CAN - Frame



- Frame structure
  - SOF: Start of Frame, one bit, Logic 0
  - 11-bit identifier: Priority of message
    - Lower value means higher priority
    - Also serves as node ID if RTR (see below) is 0
  - RTR: Remote Transmission Request, 0 if designated node needs to reply information
  - IDE: IDentifier Extension, 0 if no extension of identifier is used
  - r0: Reserved bit
  - DLC: Data Length Code, 4 bits, the number of bytes being transmitted
    - Nine possible values: 0, 1, 2, ..., 8
  - Data: Up to 8 bytes
  - CRC: Cyclic Redundancy Check (CRC) checksum on data, 16 bits
    - Includes one bit of delimiter

Diagram source: TI Application Report: Introduction to the Controller Area Network (CAN)

CAN - Frame (cont'd)



- Frame structure (cont'd)
  - ACK: Acknowledge, 2 bits
    - Nodes set first bit to dominant (Logic 0) if no error is detected in the message
    - Second bit is a delimiter, must be 1
  - EOF: End of Frame, 7 bits, set to all 1's
    - Transmitter monitors those bits and retransmits if they are not all 1's
  - IFS: InterFrame Space, 7 bits, set to all 1's
- To ensure correct sampling up to the last bit, nodes re-synchronize data sampling clock throughout the entire frame
  - To avoid large accumulation error due to small differences of sampling clocks in various modules
  - Done on each 1-to-0 bit transition

Diagram source: TI Application Report: Introduction to the Controller Area Network (CAN)

CAN - Arbitration

- Arbitration
  - Relies on the scheme that outputting a 0 will force a 0 on the bus
  - Each attempting master starts sending out the frame as if it's the bus master
  - Compares what it drives bus and what it reads from it
  - Whoever sees a mismatch stops driving the bus and just listens
  - Winning master goes on to complete its transaction
    - No corruption on the message by losing nodes
  - Higher-priority messages (likely with more zeros) get transmitted first

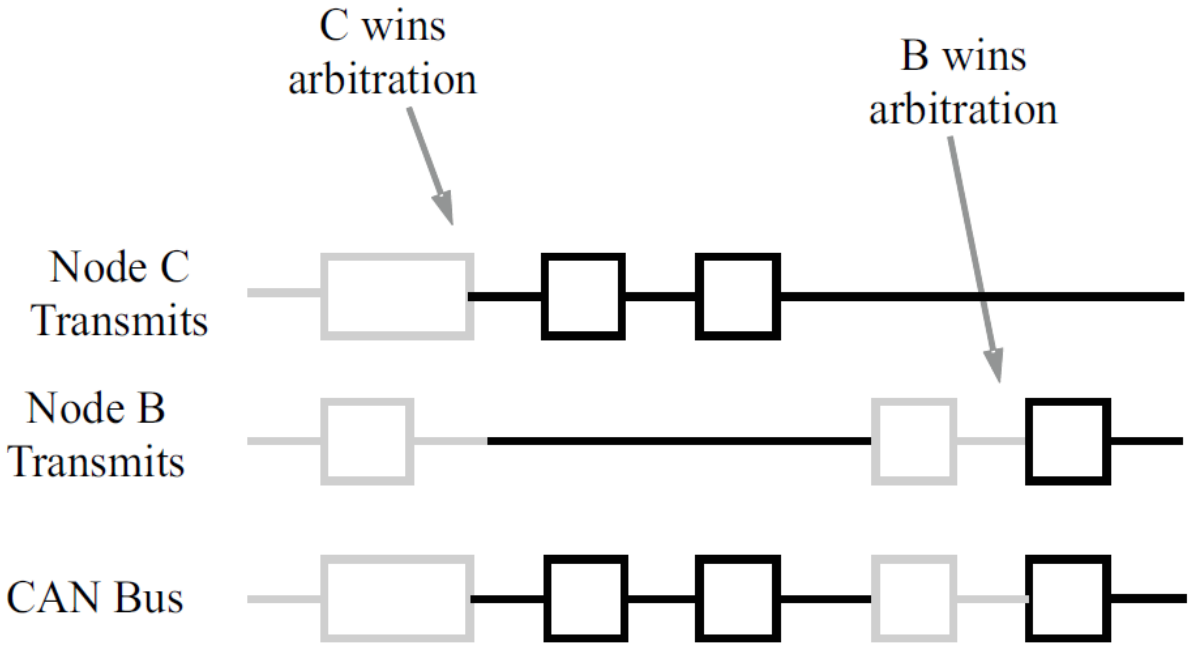


Diagram source: TI Application Report: Introduction to the Controller Area Network (CAN)

CAN – Data Sampling

- Bit sampling (when to read in a data bit)
  - A bit time slot is divided into 4 segments
    - SYNC_SEG: Synchronization segment, used for synchronization when bit changes from 1 to 0
    - PROP_SEG: Propagation segment, extra time used to compensate for the propagation delays within the network due to the line and electronics
    - PHASE_SEG1 and PHASE_SEG2: Phase segments, time buffers used to compensate for errors in the position of the beginning bit edge
  - Sample point: Point of time at which the bus level is read into bit value

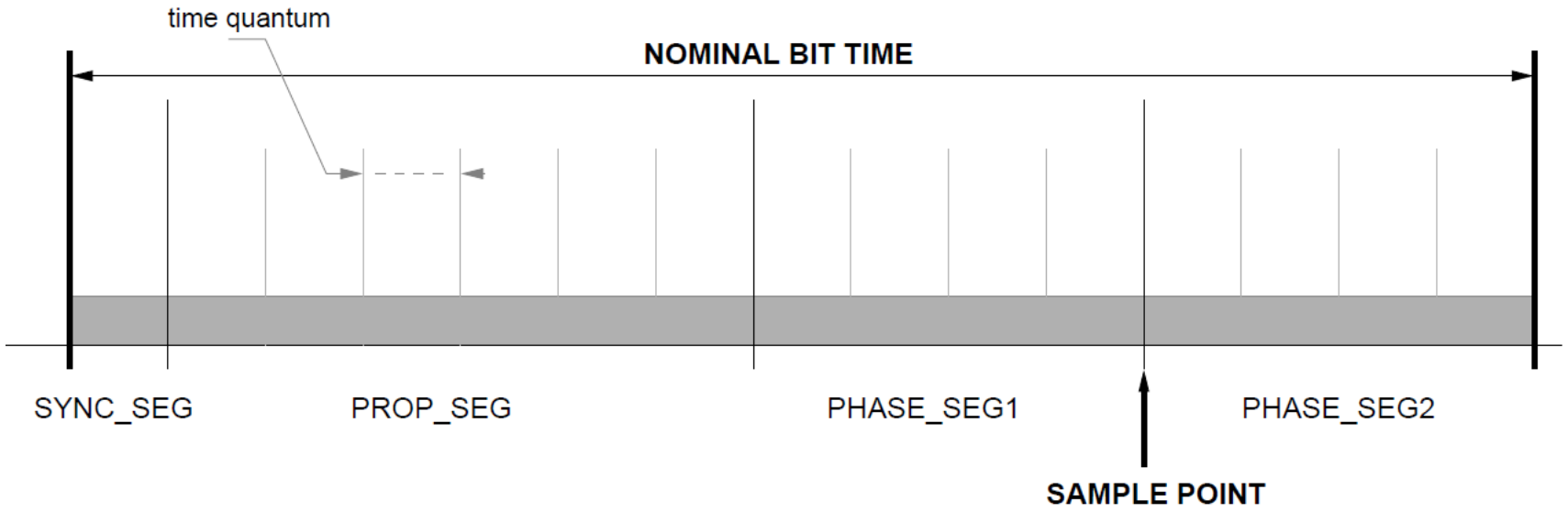


Diagram source: http://inst.cs.berkeley.edu/~ee249/fa08/Lectures/handout_canbus2.pdf