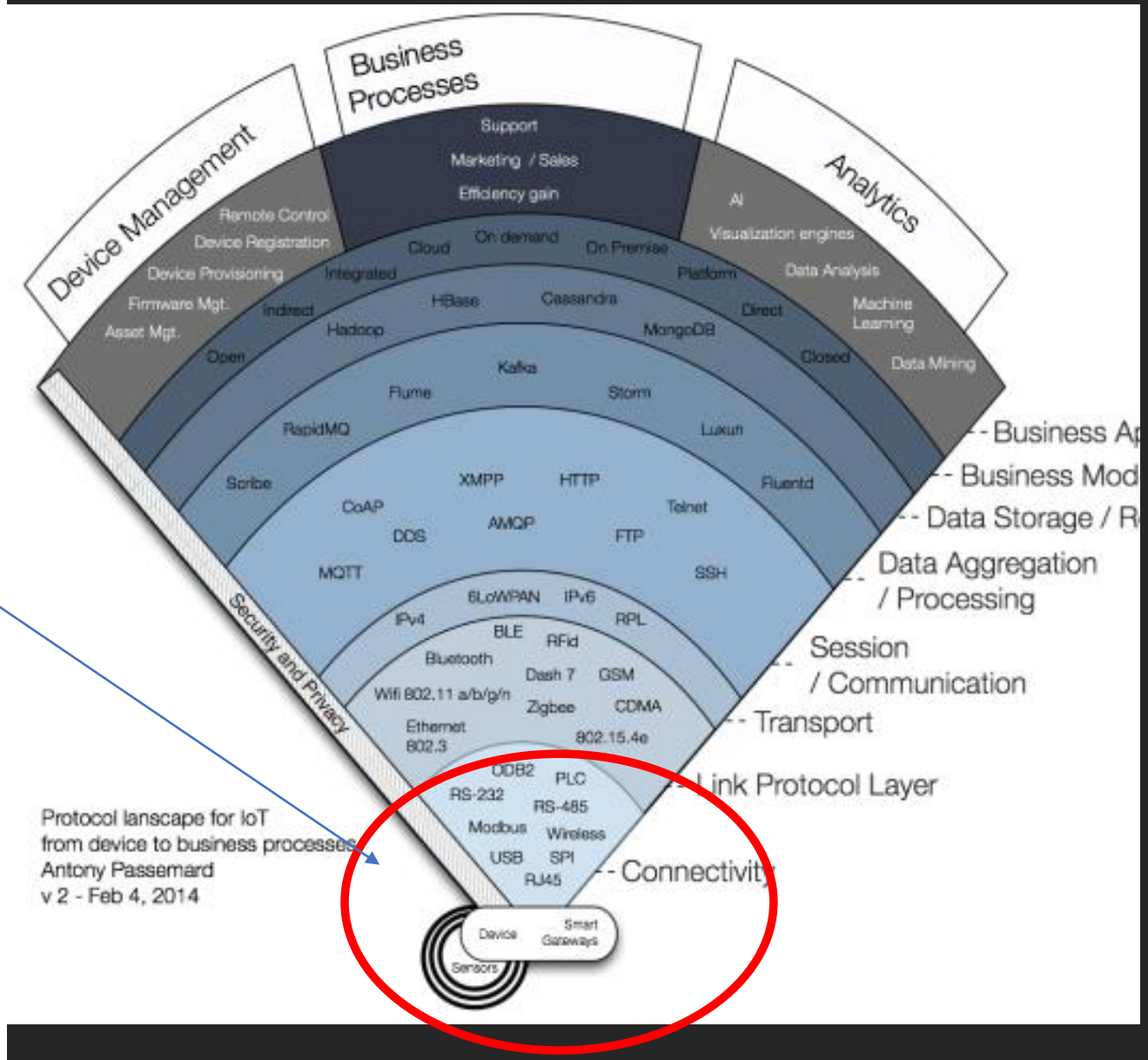
Overview of Serial Bus Protocols

Frank Walsh

We’re focussing around here…



Serial Bus

- A communication system that transfers data between components inside a computer or between computers
- Generally **2** types
  - Parallel
  - Serial
- Parallel is simplest to implement but takes up a lot of hardware ‘real estate’
- Serial requires fewer lines to transmit data but this adds complexity.
  - Synchronous – uses clock
  - Asynchronous – no clock but speed (baud rate) agreed before transmission.

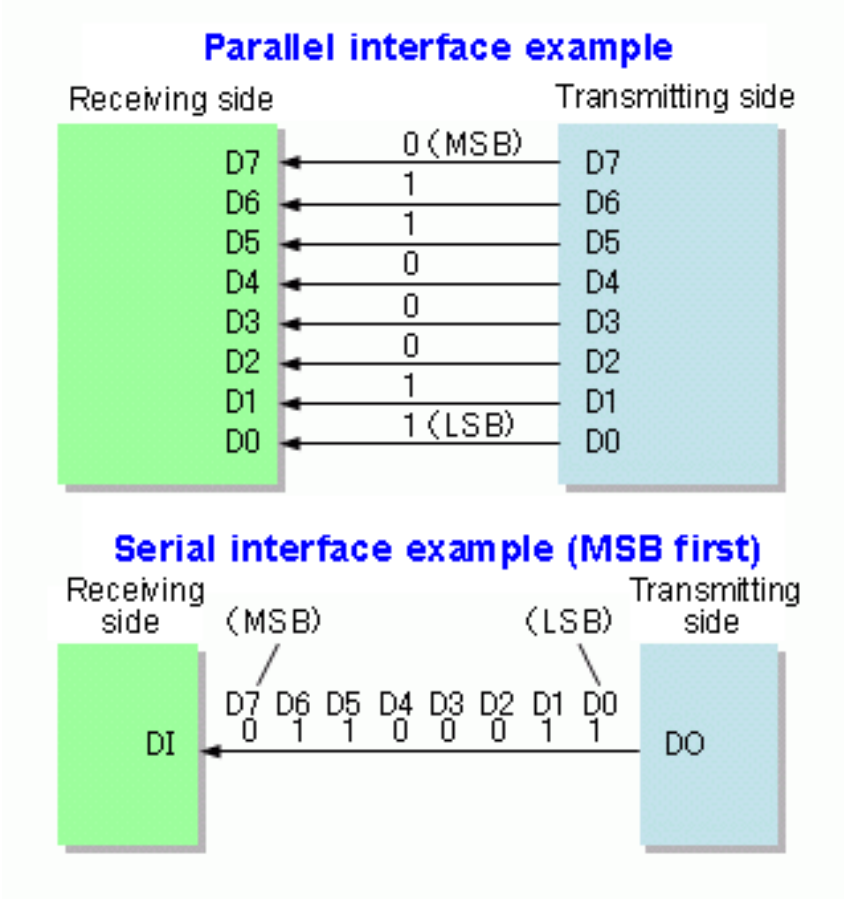
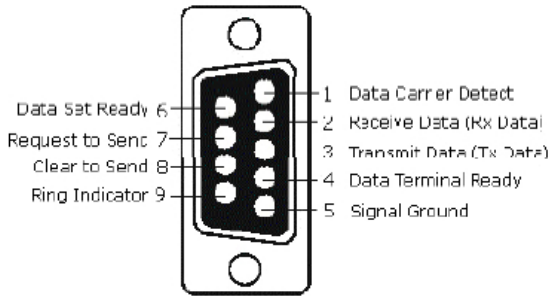


image: https://en.wikipedia.org/wiki/Parallel_communication#/media/File:Parallel_and_Serial_Transmission.gif

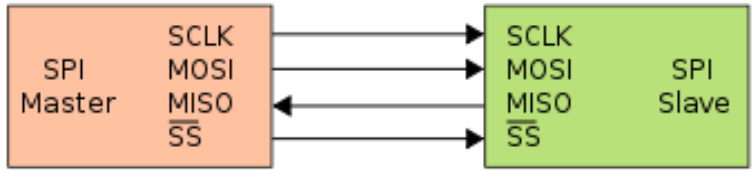


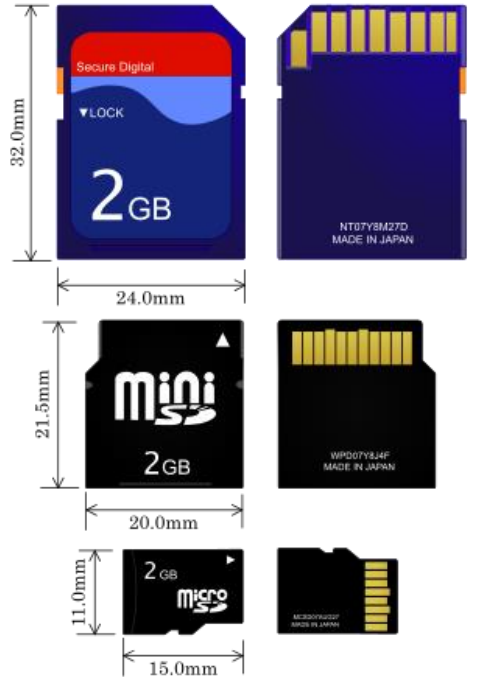
Asynchronous Serial (RS232)

- Used for 1-to-1 communication
- Many variants, simplest just uses **2** lines.
- Often used for console access to configure Network Routers

Serial Peripheral Interface (SPI)

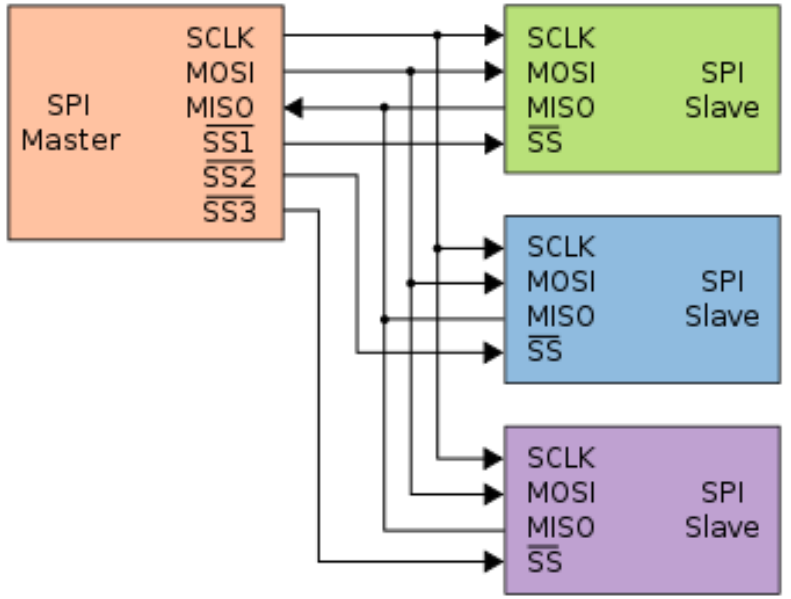
- Master and Slave Devices
  - One master and multiple slaves
- Used in liquid crystal displays and SD Cards
- Master set the speed
- Signals
  - SCLK: Serial Clock (output from master).
  - MOSI: Master Output Slave Input, or Master Out Slave In (data output from master).
  - MISO: Master Input Slave Output, or Master In Slave Out (data output from slave).
  - SS: Slave Select (pulling line low selects slave, output from master).





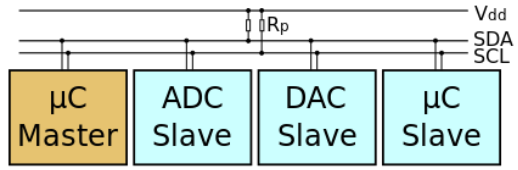
Serial Peripheral Interface (SPI)

- Slave select line goes low to select slave
- Full duplex data transmission occurs.
  - The master sends a bit on the MOSI line and the slave reads it
  - The slave sends a bit on the MISO line and the master reads it.



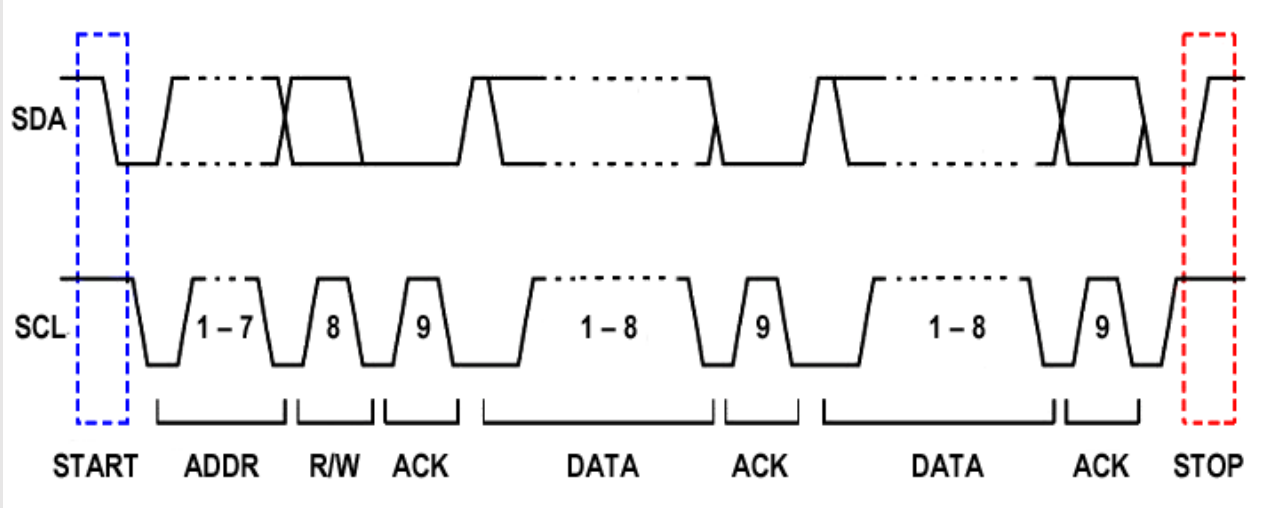
I²C (Inter-Integrated Circuit)

- Also referred to as 2-wire bus.
  - Clock(SCL) and Data(SDL)
- Used for connecting lower-speed devices to processors and microcontrollers
- Master-slave approach.
- Unlike SPI, uses addressing instead of physical Slave Select lines (hence only **2** wires).
- Speeds: 100kbs, 400kbs, 1Mbs and 3.4Mbs



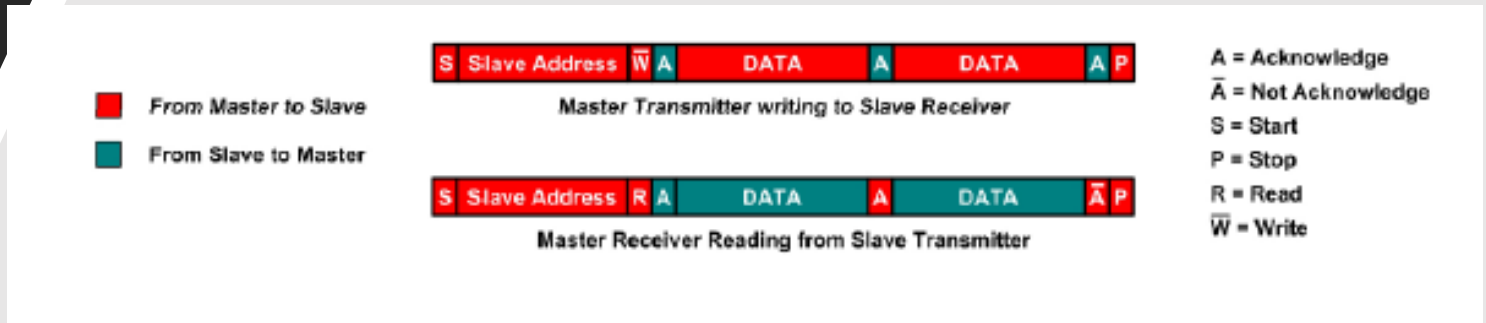
I²C (Inter-Integrated Circuit)

1. SDA,SLC start high
2. Master: SDA to low to signal start
3. Master: Send SCL with 7 bit address followed by 0 (for write)
4. Slave: pulls SDA to low for Acknowledgement
5. Master: sends **8** bit data on SDA
6. Slave: Acknowledgement
7. All: allow SDA, whin SCL is high to Stop



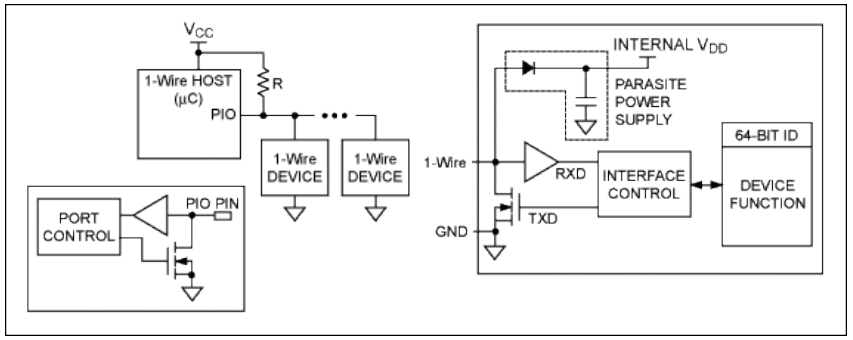
I²C (Inter-Integrated Circuit)

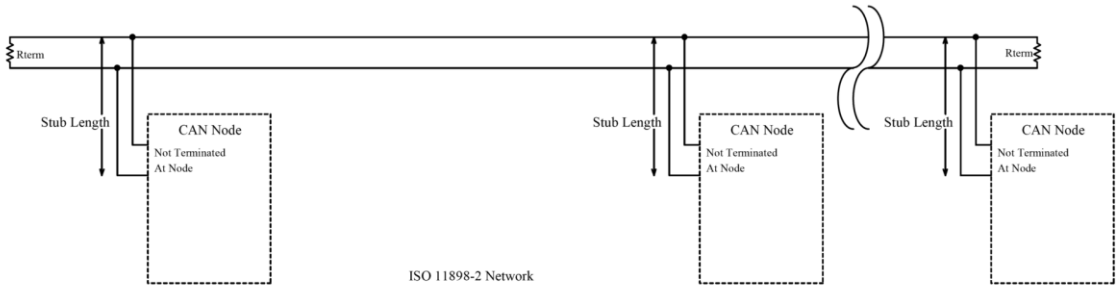
- You can transfer multiple bytes consecutively



1-Wire

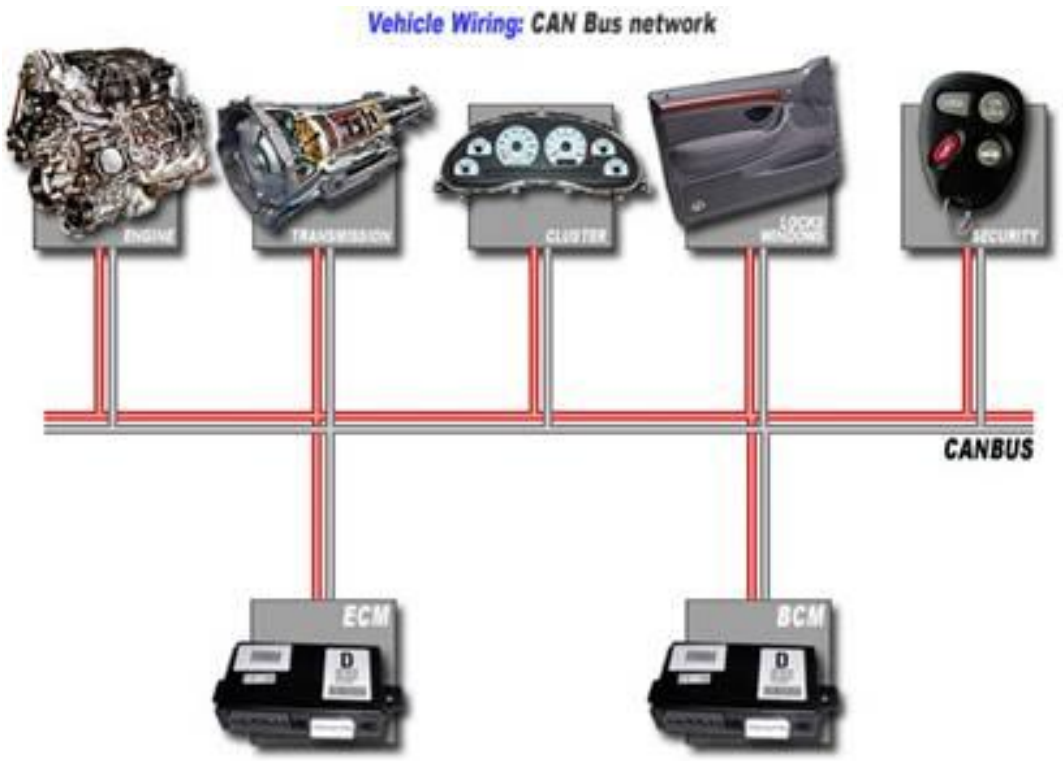
- provides low-speed data, signaling, and power over a single wire.
- Similar to I^2C, but lower data rates and longer range.
- Despite the name, you need **2** wires:
  - Data and ground.
- Because there's no power(Vcc line), 1-Wire devices use capacitors to store power
  - Parasitic device – takes power from bus.
  - power the device when the data line is active





CAN (Controller Area Network)

- A Controller Area Network (**CAN bus**) allows microcontrollers and devices to communicate with each other.
- Predominantly used in Automotive
  - Also in aviation/industrial
- No Master-Slave, it’s mutli-master
  - Any node can initiate comms.
- All nodes are connected to each other through a two wire bus



CAN (Controller Area Network)

- CAN reduces wiring requirements
- Robust protocol with built in fault tolerance
- Reliable
  - That's why it's the defacto protocol in automotive
- Relatively straight forward protocol to understand….

Universal Serial Bus (USB)

- USB resulted from mixture of connection methods used on PCs
  - Serial ports (modems)
  - Parallel ports (printers)
  - PCI (keyboards and Mice)
- Now it's "defacto"
- Low speed
  - Mice, keyboards
- Full speed
  - Other devices
- High speed
  - USB 2.0, media devices
- USB **3.0…**
- Used for all sorts of stuff!!!

USB 1.0	January 1996	Low Speed (1.5 Mbit/s)
USB 1.1	August 1998	Full Speed (12 Mbit/s)[24]
USB 2.0	April 2000	High Speed (480 Mbit/s)
USB 3.0	November 2008	SuperSpeed (5 Gbit/s)
USB 3.1	July 2013	SuperSpeed+ (10 Gbit/s)
USB 3.2	?-September 2017	SuperSpeed++ (20 Gbit/s)