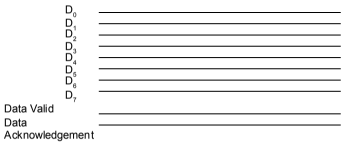
Data Communications

- The first key difference between different data communication techniques is *parallel communications* versus *serial communications*.
  - Parallel communications
    - Several bits are transferred at the same time. Typically a whole byte or word is transferred.
    - High speed transfer of data
      - Data transfer within the PC, Hard Disk ↔ CPU



Data Valid signal is high when the Data Bus has a valid value.
Data Acknowledgement signal is high to signal data has been received and the next byte can be sent.

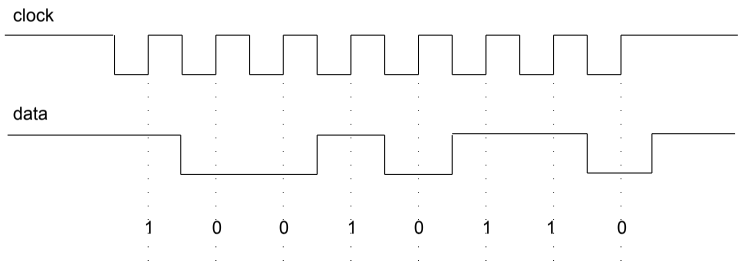
Serial Communications

- One problem with parallel communications is that it requires a lot of signal (wires).
- Serial communications is where data is transmitted one bit at a time.
  - If a byte has to be transmitted it has to be converted into a series of bits. This process is called *serialisation*.
  - The reverse process, *deserialisation*, converts a series of bits into a byte.
    - Shift registers perform this process.
- Serial communications requires very few signals!

Serial Communications (2)

- There are two types of serial communications.
  - Synchronous
    - The transfer of data is controlled by a clock signal.
  - Asynchronous
    - No clock signal.
- With synchronous serial communications each bit is transferred in step with the clock.
  - Usually with the rising edge of the clock signal.

Serial Communications (3)



Data is transferred least significant bit first.
Received data byte is 01101001 = 0x69

Serial Communications (4)

- A popular synchronous serial system is called the Serial Peripheral Interface (SPI) bus.
  - Consists of a master device and a slave device
  - SPI has four signals
    - Serial Clock (SCLK)
    - Slave Select (SS)
      - If high the output is from the slave device. If low the output is from the master device.
    - Master Output Slave Input (MOSI)
      - Output from master device.
    - Master Input Slave Output (MISO)
      - Output from slave device.

Serial Communications (5)

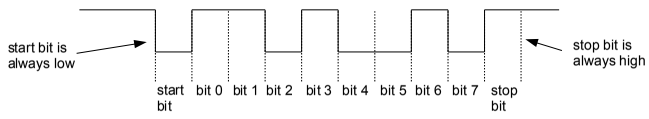
- SPI
  - Advantages
    - Full duplex (two-way) communications
    - Less signals than parallel communications.
  - Disadvantages
    - Only works over short distances.
    - No flow control or acknowledgements
      - Master could be transmitting when the slave is not ready.
  - Some applications
    - MMC and SD cards
    - Camera lenses (Canon EF mount)
    - LCDs (liquid crystal displays)

Serial Communications (6)

- Asynchronous serial communications.
  - In asynchronous serial communications there is no shared clock signal. The transmitter and receiver have their own separate clock.
  - Communications can be done over a single wire within a device.
    - When communications is between two devices, two additional wires are required so both devices have a shared view of the voltage that represents 0 and the voltage that represents 1.
  - Simple and least expensive to implement.
    - Used in many PC devices (e.g. Mice, keyboards, etc.)

Serial Communications (7)

- Since communication can occur at any time and the local clocks at the transmitting device and the receiving device may not be running in sync, the data byte being transferred needs to be framed by:
  - A ***start bit*** to tell the receiver that the first bit of the byte is about to arrive.
  - A ***stop bit*** to tell the receiver that the byte has been transferred.
    - Different systems may use 1, 1.5 or 2 stop bits.



Serial Communications (8)

- Both the transmitter and receiver have to be set for the same transmission speed.
  - Transmission speed is measured in symbol per second, ***Baud***. In most systems this is equivalent to bits per second.
- Some systems add a parity bit between the last data bit and the stop bit.
  - The parity bit is to detect data corruption.
  - There are two types, even parity and odd parity.
  - In even parity the parity bit is “1” if there is an odd number of “1” bits in the byte.
  - In odd parity the parity bit is “1” if there is an even number of “1” bits in the byte.

Serial Communications (9)

- Both the receiver and transmitter must agree on all the options for serial communications to work.
  - Same number of data bits (7 or 8)
  - Same speed
  - Same number of stop bits
  - Same parity (even or odd)

Flow Control

- When two devices transfer data to each other, this transfer can be:
  - Half-duplex
    - Both devices can transmit but only one device transmits at a time.
  - Full-duplex
    - Both devices can transmit at the same time.
- In order to prevent a device from having to receive data when it is not ready, flow control systems are implemented.

Flow Control (2)

- Typically a receiver stored received data in a buffer. When the buffer is nearly full the receiver needs to tell transmitter to stop send data until it has processed the data in the buffer.
- Software Flow Control
  - Does not require any additional signals.
  - Receiver send an XOFF code, 0x13, to the transmitter.
  - Transmitter does not send any more data until it receives an XON code, 0x11, from the receiver.

Flow Control (3)

- Advantages:
  - No extra signals required.
  - Implemented in software so no additional hardware (and cost) is required.
- Disadvantages:
  - Sending and processing the XOFF and XON codes takes time. It is not instantaneous.
  - You cannot use the XON and XOFF codes as part of the data being transmitted.

Flow Control (4)

- Hardware Flow Control
  - Common control lines are:
    - RTS (Request To Send)
    - CTS (Clear To Send)
    - DSR (Data Set Ready)
    - DTR (Data Terminal Ready).
  - The transmitting device sets its RTS signal, which signals the opposite end (the slave end such as a DCE) to begin monitoring its data input line.
  - When ready for data, the receiver sets its CTS signal.

Flow Control (5)

- If either end needs to stop the data, it lowers its respective line.
- For session-based data transfers, the DTR and DSR signals are set for the entire session (say a dialup internet call), and RTS and CTS are set for each block of data.
- Advantages:
  - Immediate effect.
- Disadvantages:
  - Additional hardware and cost.

Universal Serial Bus (USB)

- A USB system has an asymmetric design, consisting of:
  - a host,
  - a multitude of downstream USB ports, and
  - multiple peripheral devices connected in a tiered-star topology.
- Additional USB hubs may be included in the tiers, allowing branching into a tree structure with up to five tier levels.

USB (2)

- A USB host may have multiple host controllers and each host controller may provide one or more USB ports.
  - Up to 127 devices, including the hub devices, may be connected to a single host controller.
- USB devices are linked in series through hubs.
  - There always exists one hub known as the root hub, which is built in to the host controller.

USB (3)

- A physical USB device may consist of several logical sub-devices.
  - These are referred to as device functions.
  - A single device may provide several functions:
    - a webcam (video device function) with a built-in microphone (audio device function).
- USB endpoints actually reside on the connected device: the channels to the host are referred to as pipes.

USB (3)

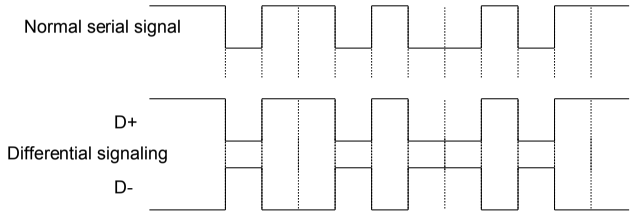
- USB device communication is based on pipes (logical channels). Pipes are connections from the host controller to a logical entity on the device named an endpoint.
  - A USB device can have up to 32 active pipes, 16 into the host controller and 16 out of the controller.
- Each endpoint can transfer data in one direction only, either into or out of the device, so each pipe is uni-directional.

USB (4)

- USB supports three data rates:
  - The Full Speed rate of 12 Mbit/s (1.5 MB/s). All USB hubs support Full Speed.
  - A Low Speed rate of 1.5 Mbit/s (187.5 kB/s).
    - It is intended primarily to save cost in low-speed devices such as keyboards, mice, and joysticks.
  - A High-Speed (USB 2.0) rate of 480 Mbit/s (60 MB/s).
    - All high-speed devices are capable of falling back to full-speed operation if necessary.

USB (5)

- USB signals are transmitted on a twisted pair data cable labeled D+ and D−. These collectively use half-duplex differential signaling to combat the effects of electromagnetic noise on longer lines.



USB (6)

- When a USB device is first connected to a USB host, the USB device enumeration process is started.
- The enumeration starts by sending a reset signal to the USB device.
  - The speed of the USB device is determined during the reset signaling.
- After reset, the USB device's information is read by the host, then the device is assigned a unique 7-bit address.

USB (7)

- If the device is supported by the host, the device drivers needed for communicating with the device are loaded and the device is set to a configured state.
- If the USB host is restarted, the enumeration process is repeated for all connected devices.
- The host controller polls the bus for traffic, usually in a round-robin fashion, so no USB device can transfer any data on the bus without an explicit request from the host controller.

USB Packets

- USB communication takes the form of packets.
- Initially, all packets are sent from the host, via the root hub and possibly more hubs, to devices.
- Some of those packets direct a device to send some packets in reply.
- Packets come in three basic types:
  - Handshake packets
  - Token packets
  - Data packets

USB Packets (2)

- Handshake packets:
  - Handshake packets are generally sent in response to data packets.
  - There are three basic types:
    - **ACK**, indicating that data was successfully received,
    - **NAK**, indicating that the data cannot be received at this time and should be retried, and
    - **STALL**, indicating that the device has an error and some corrective action (such as device initialization) should be performed.

USB Packets (3)

- Token packets
  - Tokens are only sent by the host, never a device.
  - IN and OUT tokens contain:
    - a 7-bit device number
    - 4-bit function number (for multifunction devices) and
    - command the device to transmit DATAx packets, or receive the following DATAx packets, respectively.

USB Packets (4)

- An IN token expects a response from a device.
- The response may be a NAK or STALL response, or a DATAx frame. In the latter case, the host issues an ACK handshake if appropriate.
- An OUT token is followed immediately by a DATAx frame.
- The device responds with ACK, NAK, or STALL, as appropriate.

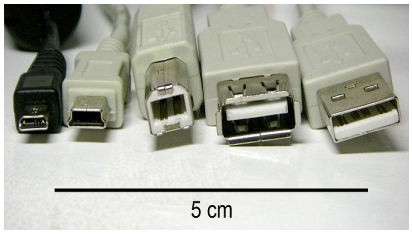
USB Packets (5)

- Data packets
  - There are two basic data packets, DATA0 and DATA1.
  - Both consist of:
    - a DATAx PID field,
    - 0–1023 bytes of data payload (up to 1024 in high speed, at most 8 at low speed),
    - and a 16-bit CRC.
  - They must always be preceded by an address token, and are usually followed by a handshake token from the receiver back to the transmitter.

USB Packet (6)

- The two packet types provide the 1-bit sequence number. If a USB host does not receive a response (such as an ACK) for data it has transmitted, it does not know if the data was received or not.
- To solve this problem, the device keeps track of the type of DATAx packet it last accepted. If it receives another DATAx packet of the same type, it is acknowledged but ignored as a duplicate. Only a DATAx packet of the opposite type is actually received.

USB Connectors



Pin	Name	Colour	Description
1	VCC	red	+5 volts
2	D-	white	Data -
3	D+	green	Data +
4	ID	none	Type A/Type B
4	ID	none	Type A/Type B

Mini-type A & Mini-type B

Pin	Name	Colour	Description
1	VCC	red	+5 volts
2	D-	white	Data -
3	D+	green	Data +
4	GND	black	Ground

Normal type A & type B

Wireless Networking

- Wireless networking, also known as WiFi, uses radio waves, like cell phones and TVs, to transmit data between devices.
  - A computer's wireless adapter translates data into a radio signal and transmits it using an antenna.
  - A wireless router receives the signal and decodes it. The router sends the information to the Internet using a physical, wired network connection.

WiFi (2)

- The radios used for WiFi communication are very similar to the radios used for cell phones. They can transmit and receive radio waves, and they can convert 1s and 0s into radio waves and convert the radio waves back into 1s and 0s.
- But WiFi radios transmit at frequencies that are considerably higher than the frequencies used for cell phones, walkie-talkies and televisions. The higher frequency allows the signal to carry more data.

WiFi (3)

- As long as they all have wireless adapters, several devices can use one router to connect to the Internet.
- This connection is convenient and fairly reliable.
  - However, if the router fails or if too many people try to use high-bandwidth applications at the same time, users can experience interference or lose their connections.

802.11 standards

- WiFi uses the 802.11 networking standards, which come in several flavours:
  - 802.11a transmits at 5 GHz and can move up to 54 megabits of data per second.
  - 802.11b is the slowest and least expensive standard. 802.11b transmits in the 2.4 GHz frequency band of the radio spectrum. It can handle up to 11 megabits of data per second.
    - For a while, its cost made it popular, but now it's becoming less common as faster standards become less expensive.

802.11 (2)

- 802.11g transmits at 2.4 GHz like 802.11b, but it's a lot faster. It can handle up to 54 megabits of data per second.
  - 802.11g is faster because it uses the same coding as 802.11a.
- 802.11n is the newest standard and is significantly improves speed and range.
  - Although theoretically capable of 54 megabits of data per second, it only achieves real-world speeds of about 24 megabits of data per second because of network congestion.

Building a Wireless Network

- If you already have several computers networked in your home, you can create a wireless network with a wireless access point.
- Otherwise you need a wireless router. This contains:
  - A port to connect to your cable or DSL modem
  - A router
  - An Ethernet hub
  - A firewall
  - A wireless access point

Building a Wireless Network (2)

- A wireless router allows you to use wireless signals or Ethernet cables to connect your computers to one another, to a printer and to the Internet.
- Most routers provide coverage for about 100 feet (30.5 meters) in all directions, although walls and doors can block the signal. The range can be extended by range extenders or repeaters.

Building a Wireless Network (3)

- When setting up a router you can specify:
  - The name of the network, known as its service set identifier (SSID).
  - The channel that the router uses.
    - Most routers use channel 6 by default. If you are experiencing interference switch to a different channel to eliminate the problem.
  - Your router's security options.
    - Many routers use a standard, publicly available sign-on, so it's a good idea to set your own username and password.

WiFi Security

- The Wired Equivalency Privacy (WEP) security measure was once the standard for WAN security.
  - But hackers discovered vulnerabilities in the WEP approach, and today it's easy to find applications and programs that can crack WEP security.
  - 128-bit WEP technology is more secure.

WiFi Security (2)

- WiFi Protected Access (WPA) is a step up from WEP and is now part of the 802.11i wireless network security protocol.
  - It uses temporal key integrity protocol (TKIP) encryption. As with WEP, WPA security involves signing on with a password.

WiFi Security (3)

- Media Access Control (MAC) address filtering is a little different from WEP or WPA.
  - It doesn't use a password to authenticate users, it uses a computer's physical hardware.
- Each computer has its own unique MAC address.
  - MAC address filtering allows only machines with specific MAC addresses to access the network. You must specify which addresses are allowed when you set up your router. This method is very secure.

WiFi Security (4)

- The system isn't foolproof.
  - A clever hacker can spoof a MAC address, i.e. copy a known MAC address to fool the network that the computer he or she is using belongs on the network.