Computer Interfaces
Serial, Parallel, GPIB, CAMAC, Oh My!

Common Implementations of Interfaces
- Parallel port (8 bits per shot)
- Serial (RS-232, RS-485)
  - usually asynchronous
- GPIB (IEEE-488) parallel
  - General Purpose Interface (or Instrument) Bus
  - originally HPIB; Hewlett Packard
- DAQ card (data acquisition)
  - like national instruments A/D, D/A, digital I/O
- CAMAC
  - Computer Automated Measurement And Control
- VME bus / VXI bus
  - modern CAMAC-like bus

A quick note on hexadecimal

<table>
<thead>
<tr>
<th>decimal value</th>
<th>binary value</th>
<th>hex value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>a</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>b</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>c</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>d</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>e</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>f</td>
</tr>
</tbody>
</table>

Hexadecimal, continued
- Once it is easy for you to recognize four bits at a time, 8 bits is trivial:
  - 01100001 is 0x61
  - 10011111 is 0x7f
- Can be handy because the ASCII code is built around hex:
  - 'A' is 0x41, 'B' is 0x42, ..., 'Z' is 0x5a
  - 'a' is 0x61, 'b' is 0x62, ..., 'z' is 0x7a
  - "A" (control-A) is 0x01, "B" is 0x02, "Z" is 0x1A
  - '0' is 0x30, '9' is 0x39
### Exchanging Data

- **Parallel**: Fast and expensive
  - Devices A, B simple, but cabling harder
  - Strobe alerts to “data valid” state

- **Serial**: Slow and cheap
  - But devices A and must convert between serial/parallel

#### Parallel Port Pinout

<table>
<thead>
<tr>
<th>Pin No (D-Type 25)</th>
<th>Pin No (Centronics)</th>
<th>SPP Signal</th>
<th>Direction</th>
<th>Register</th>
<th>Hardware Inverted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>nStrobe</td>
<td>In/Out</td>
<td>Control</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Data 0</td>
<td>Out</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>Data 1</td>
<td>Out</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Data 2</td>
<td>Out</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>Data 3</td>
<td>Out</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>Data 4</td>
<td>Out</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Data 5</td>
<td>Out</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Data 6</td>
<td>Out</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>Data 7</td>
<td>Out</td>
<td>Data</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>aAck</td>
<td>In</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>Busy</td>
<td>In</td>
<td>Status</td>
<td>Yes</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>Paper/Out</td>
<td>In</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>Select</td>
<td>In</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>nAuto-Load</td>
<td>In/Out</td>
<td>Control</td>
<td>Yes</td>
</tr>
<tr>
<td>15</td>
<td>32</td>
<td>nCarrier/nFault</td>
<td>In</td>
<td>Status</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>31</td>
<td>nDatastart</td>
<td>In/Out</td>
<td>Control</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>36</td>
<td>nSelect/Printer Select In/Out</td>
<td>Control</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>18 - 25</td>
<td>19-30</td>
<td>Ground</td>
<td>Out</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Parallel Port Access

- Most PCs have a DB-25 female connector for the parallel port
- Usually at memory address 0x378
- Windows 98 and before were easy to talk to
  - But after this, a hardware-abstraction layer (HAL) which makes access more difficult
  - One option is to fool computer into thinking you’re talking to a normal LPT (printer) device
  - Involves tying pins 11 and 12 to ground
- Straightforward on Linux
  - Direct access to all pins

### The Parallel Port

- Primarily a printer port on the PC
  - Goes by name LPTx: line printer
  - Usually LPT1
- 8 data bits
  - With strobe to signal valid data
  - Can be fast (1 Mbit/sec)

Other control and status bits for (printer) communication

---

*slide courtesy E. Michelsen*
Serial Communications

- Most PCs have a DB9 male plug for RS-232 serial asynchronous communications
  - we'll get to these definitions later
  - often COM1 on a PC
- In most cases, it is sufficient to use a 2- or 3-wire connection
  - ground (pin 5) and either or both receive and transmit (pins 2 and 3)
- Other controls available, but seldom used
- Data transmitted one bit at a time, with protocols establishing how one represents data
- Slow-ish (most common is 9600 bits/sec)

Time Is the Essence

- With separate clock and data, the transmitter gives the receiver timing on one signal, and data on another
- Requires two signals (clock and data): can be expensive
- Data values are arbitrary (no restrictions)
- Used by local interfaces: V.35, (synchronous) EIA-232, HSSI, etc.
- As distance and/or speed increase, clock/data skew destroys timing

Interpolated sample times (bit centers)

No Clock: Do You Know Where Your Data Is?

- Most long-distance, high-speed, or cheap signaling is self timed: it has no separate clock; the receiver recovers timing from the signal itself
- Receiver knows the nominal data rate, but requires transitions in the signal to locate the bits, and interpolate to the sample points
- Two General Methods:
  - Asynchronous: data sent in short blocks called frames
  - Synchronous: continuous stream of bits
  - Receiver reads the timing continuously, to stay in sync
  - Tracking requires sufficient transition density throughout the data stream
  - Used in all DSLs, DS1 (T1), DS3, SONET, all Ethernets, etc.

Asynchronous: Up Close and Personal

- Asynchronous
  - technical term meaning "whenever I feel like it"
- Start bit is always 0. Stop bit is always 1.
- The line "idles" between bytes in the "1" state.
- This guarantees a 1 to 0 transition at the start of every byte
- After the leading edge of the start bit, if you know the data rate, you can find all the bits in the byte
Can We Talk?

ASCII "A" = 0x41
9600, 8N1

- If we agree on 4 asynchronous communication parameters:
  - Data rate: Speed at which bits are sent, in bits per seconds (bps)
  - Number of data bits: data bits in each byte; usually 8
    - odd stuff often used 7
  - Parity: An error detecting method: None, Even, Odd, Mark, Space
  - Stop bits: number of stop bits on each byte; usually 1.
    - Rarely 2 or (more rarely) 1.5; just a minimum wait time; can be indefinite

RS-232: most common implementation

- RS-232 is an electrical (physical) specification for communication
  - idle, or "mark" state is logic 1;
    - -5 to -15 V (usually about -12 V) on transmit
    - -3 to -25 V on receive
  - "space" state is logic 0;
    - +5 to +15 V (usually +12 V) on transmit
    +3 to +25 V on receive
  - the dead zone is from -3 V to +3 V (indeterminate state)
- Usually used in asynchronous mode
  - so idles at -12; start jumps to +12; stop bit at -12
  - since each packet is framed by start/stop bits, you are guaranteed a transition at start
  - parity (if used) works as follows:
    - even parity guarantees an even number of ones in the train
    - odd parity guarantees an odd number of ones in the train

GPIB (IEEE-488)

- An 8-bit parallel bus allowing up to 15 devices connected to the same computer port
  - addressing of each machine (either via menu or dip-switches) determines who's who
  - can daisy-chain connectors, each cable 2 m or less in length
- Extensive handshaking controls the bus
  - computer controls who can talk and who can listen
- Many test-and-measurement devices equipped with GPIB
  - common means of controlling an experiment: positioning detectors, measuring or setting voltages/currents, etc.
- Can be reasonably fast (1 Mbit/sec)

Data Acquisition

- A PCI-card for data acquisition is a very handy thing
- The one pictured at right (National Instruments PCI-6031E) has:
  - 84 analog inputs, 16 bit
  - 2 DACs, 16 bit analog outputs
  - 8 digital input/output
  - 100,000 samples per second
  - on-board timers, counters
- Breakout box/board recommended
**CAMAC**

- This somewhat old interface provides a "crate" into which one slides modules that perform specific tasks
  - A/D conversion
  - time-to-digital converters
  - pulse generators
  - charge measurement
  - amplifiers
  - delay generators
- Frequently used in timing experiments, like nuclear physics: catch events in detector, generate signal, measure strength, etc.
- Often the modules are highly multiplexed (16 channels per card common)

**CAMAC features**

- 16-bit (newer are 24-bit) data words
- Full command cycle in 2 μs → 8 Mbit/sec
- Look-At-Me (LAM) interrupts computer when some event happens
- Commands follow N.A.F. sequence: slot number, address, function
  - so address specific modules by name/position
  - A and F values perform tasks that are defined by module
  - A often refers to channel number on multiplexed device
  - F might indicate a read, a write, a reset, or other action

**Example Interface: APOLLO**

- APOLLO is a lunar ranging apparatus that fires 20 laser pulses per second at a selected lunar reflector, measuring the time-of-flight of photons making the round trip
- Besides the essential function of data collection and apparatus coordination, we wanted remote operation capability
- We also required strict thermal control
Catalog of APOLO Interfaces

- Uses a Linux PC (runs for a year at a time, no crashes)
- Two GPIB devices
  - GPS-disciplined clock; actuated optics (mirror tilt, lens focus)
- 5 RS-232 devices
  - Motor that spins optic (8N1 @ 57600); laser control (8E1 @ 9600); CCD camera control (8N1 @ 115200); laser power meter (bolometer) (8N1 @ 9600); GPS clock (7E1 @ 9600)
- CAMAC crate with two devices
  - TDC for 10 ps timing; custom module to control timing
  - Another device sits passively in crate, no access to data way
- DAQ card for analog input, digital output
  - Analog inputs for RTDs (temperature); flow meters; pulse energy; telescope tilt angle
  - Digital outputs for relay control; turning devices on and off
- Parallel port used for additional digital outputs for more relays

Example Temperature Record

Reading

- Read 6.7.3; skim 6.7.5; read 6.7.7; 6.7.9