Serial Communications

Prof. Stephen A. Edwards  
sedwards@cs.columbia.edu

Columbia University  
Spring 2008

Early Serial Communication

RS-232

Defined in early 1960s  
Serial, Asynchronous, Full-duplex,  
Voltage-based, point-to-point, 100 ft+ cables

\[
\begin{align*}
+12V & : \text{SPACE} = 0 \\
+3V & \\
-3V & : \text{MARK} = 1 \\
-12V & 
\end{align*}
\]

Later Serial Communication

RS-232 Signals

RS-232

Serial Communications – p.

Signaling DB-9 DTE … Meaning

<table>
<thead>
<tr>
<th>Signal</th>
<th>DCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RxD</td>
<td>2</td>
</tr>
<tr>
<td>TxD</td>
<td>3</td>
</tr>
<tr>
<td>SG</td>
<td>5</td>
</tr>
<tr>
<td>DSR</td>
<td>6</td>
</tr>
<tr>
<td>DTR</td>
<td>4</td>
</tr>
<tr>
<td>DCD</td>
<td>1</td>
</tr>
<tr>
<td>RTS</td>
<td>7</td>
</tr>
<tr>
<td>CTS</td>
<td>8</td>
</tr>
<tr>
<td>RI</td>
<td>9</td>
</tr>
</tbody>
</table>

Receiving RS-232

Most UARTs actually use $16 \times$ clocks

Baud Rate

Baud: bits per second

<table>
<thead>
<tr>
<th>Baud</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>ASR-33 Teletype</td>
</tr>
<tr>
<td>300</td>
<td>Early acoustic modems</td>
</tr>
<tr>
<td>1200</td>
<td>Direct-coupled modems c. 1980</td>
</tr>
<tr>
<td>2400</td>
<td>Modems c. 1990</td>
</tr>
<tr>
<td>9600</td>
<td>Serial terminals</td>
</tr>
<tr>
<td>19200</td>
<td></td>
</tr>
<tr>
<td>38400</td>
<td>Typical maximum</td>
</tr>
</tbody>
</table>

Physical Variants

Connectors: DB-25, DB-9, Mini DIN-8

RS-422: Differential signaling  
RS-485: Bus-like
The I²C Bus

Philips invented the Inter-IC bus c. 1980 as a very cheap way to communicate slowly among chips. E.g., good for setting control registers.

- 100, 400, and 3400 KHz bitrates

SCL: Clock, generated by a single master.
SDA: Data, controlled by either master or slaves.

I²C Bus Transaction

USB: Universal Serial Bus

1.5 Mbps, 12 Mbps, and 480 Mbps (USB 2.0)
Point-to-point, differential, twisted pair
3–5m maximum cable length

USB Connectors

USB signaling

NRZI: 0 = toggle, 1 = no change
Bit stuffing: 0 automatically inserted after six consecutive 1s

Each packet prefixed by a SYNC field: 3 0s followed by two 1s
Low- vs. full-speed devices identified by different pull-ups on D+/D- lines.

USB Packets

Always start with SYNC
Then 4-bit type, 4-bit type complemented
2 bits distinguish Token, Data, Handshake, and Special, other two bits select sub-types
Then data, depending on packet type
Data checked using a CRC
Addresses (1-128) assigned by bus master, each with 16 possible endpoints

USB Bus Protocol

Polled bus: host initiates all transfers.
Most transactions involve three packets:
  "Token" packet from host requesting data
  Data packet from target
  Acknowledge from host
Supports both streams of bytes and structured messages (e.g., control changes).

USB Data Flow Types

Control
  For configuration, etc.
Bulk Data
  Arbitrary data stream: bursty
Interrupt Data
  Timely, reliable delivery of data. Usually events.
Isochronous Data
  For streaming real-time transfer: prenegotiated bandwidth and latency

Layered Architecture

Function Layer
  Host
  Interconnect
  Physical Device

USB Device Layer
  USB Host Controller
  USB Bus Interface

Logical communications flow
  Actual communications flow
  Implementation Focus Area
**USB: Flash Card Device**

Device Descriptors:
- ID Vendor: 0x05e3, Genesys Logic, Inc.
- ID Product: 0x0760, Flash Reader
- Maximum Transfers (Configuration): 64
- Interface Descriptors:
  - ID Vendor: 0x05e3, Genesys Logic, Inc.
  - ID Product: 0x0760, Flash Reader
  - Maximum Transfers: 0x81, EP 1 IN
  - Attributes: Bulk
  - Transfer Type: Interrupt
  - Max Packet Size: 1
- Language IDs: 0409 English (US)

**USB: Mouse Device**

Device Descriptors:
- ID Vendor: 0x04b4, Cypress Semiconductor
- ID Product: 0x0001, Mouse
- Maximum Transfers (Configuration): 64
- Interface Descriptors:
  - ID Vendor: 0x04b4, Cypress Semiconductor
  - ID Product: 0x0001, Mouse
  - Maximum Transfers: 0x81, EP 1 IN
  - Attributes: Bulk
  - Transfer Type: Interrupt
  - Max Packet Size: 1
- Language IDs: 0409 English (US)

**Philips ISP1362 USB 2.0 Controller**

- On the DE2, one downstream port, one host
- Operates at 12 or 480 Mbps speeds
- Two control endpoints + 14 user endpoints
- 4096 (host) + 2462 (device) bytes buffer memory
- Supports DMA data transfers
- Many configuration and status registers
- 150-page data “sheet” + 99-page embedded programming guide