Universal Asynchronous Receiver Transmitter

Networks and Embedded Systems
Second Grade Level
Wolfgang Neff
UART (1)

• Design
  – Serial Communication Protocol
  – Point-to-Point Communication
  – Unidirectional Transmission Lines
    • RxD: Received Data
    • TxD: Transmitted Data
  – Hardware flow control (optional)
    • RTS: Request To Send
    • CTS: Clear To Send
UART (2)

• Quite a lot of configuration
  – Baud rate (bps: bits per second)
  – Number of data bits (Baudot: 5 bit, ASCII: 7 bit)
  – Parity mode (even, odd, none)
  – Number of stop bits (one, two)
  – Example: 9600/8N1
    • 9600 bits per second (104 μs per bit)
    • 8 data bits, no parity bit, 1 stop bit
What is the parity bit?

- Simplest form of an error detecting code
- Two variants of parity bit: even (E) or odd (O)
- Number of 1s including parity bit must be E or O

<table>
<thead>
<tr>
<th>Data Bits</th>
<th>Even Parity</th>
<th>Odd Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>+ 0 = E</td>
<td>+ 1 = O</td>
</tr>
<tr>
<td>1010001</td>
<td>+ 1 = E</td>
<td>+ 0 = O</td>
</tr>
<tr>
<td>1101001</td>
<td>+ 0 = E</td>
<td>+ 1 = O</td>
</tr>
<tr>
<td>1111111</td>
<td>+ 1 = E</td>
<td>+ 0 = O</td>
</tr>
</tbody>
</table>
UART (4)

• UART means asynchronous transmission
  – Three line connection
  – Bit transmission
    • Example: 8N1, G = \text{47}_{\text{hex}} = 01000111

<table>
<thead>
<tr>
<th>standby</th>
<th>sync</th>
<th>data bits</th>
<th>sync</th>
<th>standby</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>0</td>
<td>1 2 3 4 5 6 7</td>
<td>stop</td>
<td></td>
</tr>
</tbody>
</table>
RS-232 (1)

- How is it designed?
  - Based on UART
  - Additional control lines
    - RI: Ring Indicator
    - DTR: Data Terminal Ready
  - Flow Control
    - No handshaking
    - Hardware handshaking (RTS and CTS)
    - Software handshaking (XON and XOFF control characters)
How is it designed? (continued)

- Voltage Levels
  - 0: +3 ... +15 V (space), UART: GND (0 V)
  - 1: -15 ... -3 V (mark), UART: VCC (5.0 V, 3.3 V)

- Connectors
  - 25-pin D-subminiature connector (standard recommendation)
  - 9-pin D-subminiature connector (widely used)
How does flow control work?

**Hardware handshake**

- GND
- TxD
- RxD
- RTS
- CTS

Device 1

- RTS: ready to receive data
- CTS: request to send data

Device 2

**Software handshake**

- Buffer full
- High water mark
- Overflow

- Low water mark
- Buffer empty
- Underflow

- Buffer full
- High water mark
- XON: sent if below low water mark

- Low water mark
- Buffer empty

- Underflow

- Buffer full
- High water mark
- XOFF: sent if above high water mark
RS-232 (4)

• How data is transmitted
  – Example: 8N1, \( G = 47_{\text{hex}} = 01000111 \)

<table>
<thead>
<tr>
<th>Standby</th>
<th>Sync</th>
<th>Data Bits</th>
<th>Parity</th>
<th>Sync</th>
<th>Standby</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0 1 2 3 4 5 6 7</td>
<td>E</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Logical: 0, 1  
Signal: -15 V ... +15 V
RS-232 (5)

- Real-Life Example (3.3 V positive logic levels)
RS-232 (6)

• Advantages
  – Simplicity
  – Low cost
  – Easy to implement
  – Widely used
  – Converters and adaptors available
• Disadvantages
  – Point-to-point
  – No automatic configuration
  – Many configuration settings
  – Requires transceiver chip
    • MAX233 level shifter