Serial Interfaces – Part 1

ECE 153B
Sensor & Peripheral Interface Design
Winter 2016
Serial Interfaces

“Simple” Serial Interfaces

- RS-232C (UART)
  - Provides for point-to-point communications, primarily
    - Among the simplest serial communication methods
    - Generally no address decoding, .... though “multi-drop” systems exist
  - Max speed ~ 1 M bits per sec

- I^2^C (Inter–Integrated Circuit) Bus
  - Serial, 2-wire (plus ground) bus
    - Note that it is a true bus (as opposed to point-to-point communications)
  - Popular in embedded systems

- SPI (Serial Peripheral Interface ) Bus
  - Also referred to as SSI (Synchronous Serial Interface)
  - Serial, 4-wire bus
  - Operates in full duplex mode
Serial Interfaces

“Complex” Serial Interfaces

- USB
  - Universal Serial Bus
  - PC to peripheral connection
  - Complex protocols (hardware and software)

- Firewire
  - High data rate communication between electronic devices
  - Developed by Apple & Sony, standardized as IEEE 1394
  - Complex protocols (hardware and software)
Serial Interfaces

“Complex” Serial Interfaces (continued)

- Ethernet
  - Local Area Networking
    - computer-to-computer linkage
  - Developed by Xerox PARC, IEEE 802.3
  - Relies on protocols to implement sharing of the medium
    - original versions were bus-like, now most common is point-to-point
Serial Interfaces

- Wireless Serial Interfaces
  - Bluetooth
    - Wireless (radio) communication between electronic devices
    - Protocols are based on ethernet-like model
    - RF portion quite complex
  - WiFi (IEEE 802.11)
    - Wireless LAN
      - Computer-to-computer connections
    - Same protocols as ethernet, plus more
    - RF portion (as in the case of Bluetooth) quite complex
Serial Communication – Signaling

- RS-232C (UART data input/output)
- Start bit, Data bits, Parity bit and Stop bit(s)
Serial Communication

- Start bit and Data bits
  - Once the start bit has been sent, the transmitter sends the actual data bits
  - There may either be 5, 6, 7 or 8 data bits
  - Both receiver and transmitter must know and agree on the number of data bits, as well as the baud rate
    - Simple protocol
    - Almost all devices transmit data using either 7 or 8 data bits
Serial Communication

- Stop bit
  - After the data has been transmitted, a stop bit is sent
  - A stop bit has a value of 1 (a mark state) and it can be detected correctly even if the previous data bit also had a value of 1
    - This is accomplished by the stop bit's duration
    - Stop bits can be 1, 1.5, or 2 bit periods in length.
Serial Communication

- Parity Bit
  - A parity bit affords a small amount of error checking
    - Detects data corruption that might occur during transmission
    - Even parity, odd parity or no parity can be used
      - When even or odd parity is being used, the number of logical 1 bits in each data byte are counted, and a single bit is transmitted following the data bits to indicate whether the total number of 1 bits just sent (including the parity bit) is even or odd
Serial Communication

- Parity error checking is very rudimentary

  - While it can tell you if there is a single bit error in the character, it doesn't show which bit was received in error

    - Parity is an example of a single bit, error detection code

  - Note that if an even number of bits are in error (e.g. 2) then the parity bit would not indicate any error at all!
Bi-DIRECTIONAL, SERIAL COMMUNICATIONS

- Full-duplex
  - Send and receive data at the same time
    - Two-way (full duplex) communications is possible with just three wires: one to send, one to receive, and a common signal ground wire

- Half-duplex
  - Send and receive, but not simultaneously
    - Can be configured to use only two wires in the cable - the data line and the signal ground
    - Normally, some sort of ad hoc protocol would be used to avoid simultaneous transmission attempts from both ends
    - Seldom used in modern computer communications
RS-232C

- RS-232C: “Recommend Standard 232, version C”
  - C is the latest revision (EIA 1969) of the standard.
  - Serial ports on most (older) computers use a subset of the RS-232C standard
    - Full RS-232C standard specifies a 25-pin "D" connector of which 22 pins are used
  - Most of these pins are not needed for normal PC communications
    - most PCs are equipped with male D type connectors having only 9 pins
      - More on the pinout later ...
  - On modern computers with no serial port, RS-232C interface implemented with a USB to RS-232C converter
RS-232C Voltage Levels

- The RS-232 standard defines the voltage levels that correspond to logic one and logic zero levels for the data transmission and the control signal lines.

- Valid signals are either in the range of +3 to +15 volts or the range −3 to −15 volts with respect to the ground/common pin.
  - The range between −3 to +3 volts is not a valid RS-232 level.

- For data transmission lines (TxD, RxD):
  - Logic one is defined as a negative voltage and the signal condition is called "mark".
  - Logic zero is positive and the signal condition is termed "space".

- Control signals have the opposite polarity:
  - The asserted or active state is positive voltage and the deasserted or inactive state is negative voltage.
RS-232C Voltage Levels

The ASCII Character “K” (4B)
RS-232C Voltage Levels

- Because the voltage levels are bipolar and higher than logic levels typically used by integrated circuits, special intervening driver circuits are required to translate logic levels
  - Level shifter required to translate RS-232C levels to CMOS levels (i.e., 5V, 3.3V, etc.)
  - MAX211 (U208) on ECE 153B SYS Board
    - Level shifter utilizes an on chip “charge pump” to generate necessary voltages from single polarity power supply
RS-232C on ECE 153B SYS Board

![Diagram of RS-232C on ECE 153B SYS Board]

- UART
- Connector
- Level Shifter
End points: DTE and DCE Devices

- DTE: Data Terminal Equipment
- DCE: Data Communications Equipment

- Historically, this refers to the connection between a computer terminal (DTE) and a modem (DCE)

- In RS-232C, these terms are used to indicate the pin-out for the connectors on a device and the direction of the signals on the pins

- In modern practice, DTE is a computer and DCE is everything else
RS-232C Connector Standards

- DTE devices use a 25-pin male connector
  - In the 9-pin abbreviated version, a DB-9 male connector
- DCE devices use a 25-pin female connector (or DB-9/F)
  - A DTE device and a DCE device can communicate using a straight-through pin-for-pin connection
- To connect two like devices (i.e. DTE-to-DTE or DCE-to-DCE), you must use a “null modem” cable
  - Null modem cables cross the transmit and receive lines in the cable, and are discussed later
### RS-232C Connectors

A nine-pin version of the connector (the DB-9) is often used to save space.

#### 25 Pin Connector on a DTE device (PC connection)

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Direction of signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Protective Ground</td>
</tr>
<tr>
<td>2</td>
<td>Transmitted Data (TD) Outgoing Data (from a DTE to a DCE)</td>
</tr>
<tr>
<td>3</td>
<td>Received Data (RD) Incoming Data (from a DCE to a DTE)</td>
</tr>
<tr>
<td>4</td>
<td>Request To Send (RTS) Outgoing flow control signal controlled by DTE</td>
</tr>
<tr>
<td>5</td>
<td>Clear To Send (CTS) Incoming flow control signal controlled by DCE</td>
</tr>
<tr>
<td>6</td>
<td>Data Set Ready (DSR) Incoming handshaking signal controlled by DCE</td>
</tr>
<tr>
<td>7</td>
<td>Signal Ground Common reference voltage</td>
</tr>
<tr>
<td>8</td>
<td>Carrier Detect (CD) Incoming signal from a modem</td>
</tr>
<tr>
<td>9</td>
<td>Data Terminal Ready (DTR) Outgoing handshaking signal controlled by DTE</td>
</tr>
<tr>
<td>10</td>
<td>Ring Indicator (RI) Incoming signal from a modem</td>
</tr>
</tbody>
</table>

#### 9 Pin Connector on a DTE device (PC connection)

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Direction of signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Carrier Detect (CD) (from DCE) Incoming signal from a modem</td>
</tr>
<tr>
<td>2</td>
<td>Received Data (RD) Incoming Data from a DCE</td>
</tr>
<tr>
<td>3</td>
<td>Transmitted Data (TD) Outgoing Data to a DCE</td>
</tr>
<tr>
<td>4</td>
<td>Data Terminal Ready (DTR) Outgoing handshaking signal</td>
</tr>
<tr>
<td>5</td>
<td>Signal Ground Common reference voltage</td>
</tr>
<tr>
<td>6</td>
<td>Data Set Ready (DSR) Incoming handshaking signal</td>
</tr>
<tr>
<td>7</td>
<td>Request To Send (RTS) Outgoing flow control signal</td>
</tr>
<tr>
<td>8</td>
<td>Clear To Send (CTS) Incoming flow control signal</td>
</tr>
<tr>
<td>9</td>
<td>Ring Indicator (RI) (from DCE) Incoming signal from a modem</td>
</tr>
</tbody>
</table>
RS232C Data & Control Signals

<table>
<thead>
<tr>
<th>Name</th>
<th>Typical purpose</th>
<th>Abbreviation</th>
<th>Origin</th>
<th>DB-25 pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Terminal Ready</td>
<td>Indicates presence of DTE to DCE.</td>
<td>DTR</td>
<td>•</td>
<td>20</td>
</tr>
<tr>
<td>Data Carrier Detect</td>
<td>DCE is connected to the telephone line.</td>
<td>DCD</td>
<td>•</td>
<td>8</td>
</tr>
<tr>
<td>Data Set Ready</td>
<td>DCE is ready to receive commands or data.</td>
<td>DSR</td>
<td>•</td>
<td>6</td>
</tr>
<tr>
<td>Ring Indicator</td>
<td>DCE has detected an incoming ring signal on the telephone line.</td>
<td>RI</td>
<td>•</td>
<td>22</td>
</tr>
<tr>
<td>Request To Send</td>
<td>DTE requests the DCE prepare to receive data.</td>
<td>RTS</td>
<td>•</td>
<td>4</td>
</tr>
<tr>
<td>Clear To Send</td>
<td>Indicates DCE is ready to accept data.</td>
<td>CTS</td>
<td>•</td>
<td>5</td>
</tr>
<tr>
<td>Transmitted Data</td>
<td>Carries data from DTE to DCE.</td>
<td>TxD</td>
<td>•</td>
<td>2</td>
</tr>
<tr>
<td>Received Data</td>
<td>Carries data from DCE to DTE.</td>
<td>RxD</td>
<td>•</td>
<td>3</td>
</tr>
<tr>
<td>Common Ground</td>
<td></td>
<td>GND</td>
<td>common</td>
<td>7</td>
</tr>
<tr>
<td>Protective Ground</td>
<td></td>
<td>PG</td>
<td>common</td>
<td>1</td>
</tr>
</tbody>
</table>

All signal names are from the standpoint of the DTE
Hardware Flow Control

- RTS/CTS handshaking
  - RTS stands for “Request To Send”
  - CTS stands for “Clear to Send”

- This signal pair is used when hardware flow control is enabled in both the DTE and DCE devices
  - RTS originates from DTE; CTS originates from DCE

- There are at least two variations on the use of RTS/CTS in the standard
  - If you plan to use RTS/CTS, be sure to understand the exact meaning of each of these two control lines
  - RS232C spec defines source of signal but not specific usage

- Hardware flow control rarely used in modern systems
  - RxD, TxD and GND are all that are necessary for full duplex operation
Software Flow Control

- Xon/Xoff or "software" flow control
  - Software flow control uses special control characters transmitted from one device to another to tell the other device to stop or start sending data
    - DC1 (0x11) means start (transmitter on – aka “Xon”)
    - DC3 (0x13) means stop (transmitter off – aka “Xoff”)
  - With software flow control, the RTS and CTS lines are not used
Additional Handshaking Signals

- DTR stands for “Data Terminal Ready”
  - Intended function is very similar to the RTS line except it has to do with indicating readiness to start or stop a session

- DSR (“Data Set Ready”) is the companion to DTR in the same way that CTS relates to RTS

- Some serial devices use DTR and DSR as signals to simply confirm that a device is connected and is turned on
  - DTR and DSR lines were originally designed to provide an alternate method of hardware handshaking
    - It would be pointless to use both RTS/CTS and DTR/DSR for flow control signals at the same time.
    - Because of this, DTR and DSR are rarely used for flow control
Carrier Detect & Ring Indicator

- **CD (or DCD)** stands for “Carrier Detect”
  - Carrier Detect is used by a modem to signal that it has made a connection with another modem
    - i.e., that it has detected a carrier tone
  - The last remaining line is **RI (or “Ring Indicator”)**
    - A modem toggles the state of this line when an incoming call rings your phone
      - The Carrier Detect (CD) and the Ring Indicator (RI) lines are only available in connections to a modem
      - Because most modems transmit status information to a host when either a carrier signal is detected (i.e., when a connection is made to another modem) or when the line is ringing, these two lines are rarely used
Baud vs. Bits per Second

- Baud unit named after Jean Maurice Emile Baudot
  - an officer in the French Telegraph Service
  - credited with devising the first uniform-length 5-bit code for characters of the alphabet (late 19th century)

- Baud refers to the modulation rate or the number of times per second that a line changes state
Baud vs. Bits per Second

- Baud is not always the same as bits per second
  - If you connect two serial devices together using direct cables then baud and bps are in fact the same
  - Thus, if you are running at 19200 bps, then the line is also changing states at up to 19200 times per second
  - But when considering modems, this isn't always the case
    - \( bps = \text{baud per second} \times \text{the number of bits per baud} \)
Baud vs. Bits per Second

- Modems often transfer signals over a telephone line, where the baud rate is actually limited to a maximum of 2400 baud
  - a physical (audio) restriction of the lines provided by the phone company
  - the increased data throughput achieved with 9600 or higher baud-rate modems is accomplished by using sophisticated phase modulation, and data compression techniques.
Cable Lengths

- The RS-232C standard imposes a cable length limit of 50 feet
  - You can usually ignore this "standard", since a cable can be as long as 10,000 feet at baud rates up to 19200 if you use a high quality, well shielded cable and beefy line drivers
  - The external environment has a strong effect on lengths for unshielded cables
    - In electrically noisy environments, even very short cables can pick up stray signals
Null Modem Cables & Adapters

- Two like devices (DTE or DCE) connected with a straight RS232 cable
  - Won’t work
    - transmit line on each device will be connected to the transmit line on the other device and the receive lines will likewise be connected to each other
  - Unless a Null Modem cable or Null Modem adapter is used
    - this crosses the receive and transmit lines so that transmit on one end is connected to receive on the other end and vice versa
  - In addition to transmit and receive, DTR & DSR, as well as RTS & CTS are also crossed in a Null Modem connection.
Synchronous and Asynchronous Communications

- Moving forward, understand that there are two basic types of serial communications, synchronous and asynchronous.
  - With synchronous communications, the two devices initially synchronize themselves to each other, and then continually send characters to stay in sync.
  - Even when data is not really being sent, a constant flow of bits allows each device to know where the other is at any given time.
    - Each character sent is either actual data or an “idle” character.
    - SYN (0x16) is the ASCII “synchronous idle” character.
  - Clock signal does not have to be explicitly transmitted, it can be recovered at the receiver via a Clock Recovery Unit (CRU) circuit.
Synchronous and Asynchronous Communications

- Asynchronous means "no synchronization", and thus does not require sending and receiving idle characters
  
  However, the beginning and end of each byte of data must be identified by start and stop bits
  - RS-232C is an a prime example of asynchronous communication

- The start bit indicates when the data byte is about to begin and the stop bit signals when it ends
  
  - The requirement to send these additional two bits causes asynchronous communication to be slightly slower than synchronous communication at an equivalent Baud rate
  
  - However it has the advantage that the processor does not have to deal with the additional idle characters.
Synchronous and Asynchronous Communications

- Synchronous communications allows faster data transfer rates than asynchronous methods
  - because additional bits to mark the beginning and end of each data byte are not required
- Serial ports on PCs are asynchronous devices and therefore only support asynchronous serial communications
- Most processors intended to be used in embedded systems support high data rate synchronous serial communications as well
Synchronous and Asynchronous Communications

- An asynchronous line that is idle is identified with a value of 1 (also called a ‘mark’ state)
  - By using this value to indicate that no data is currently being sent, the devices are able to distinguish between an idle state and a disconnected line
  - When a character is about to be transmitted, a start bit is sent. A start bit has a value of 0 (also called a ‘space’ state)
    - Thus, when the line switches from a value of 1 to a value of 0, the receiver is alerted that a data character is about to be sent