ECE 381 - Microcontrollers

Lecture 5 – RS-232 Interface and Protocol

SOUTHERN ILLINOIS UNIVERSITY
EDWARDSVILLE

Problem: How To Send Data To/From Microcontroller

- Example: Microcontroller sampling temperature sensor, want to make data log on a computer (12-bit ADC)
- How to transmit that data?
  - In parallel or serial?
  - How is data encoded?
  - Does the micro have to sync with PC?
  - How is the data clocked?
  - What are the electrical signal levels (i.e. uC 3.3V, PC 5V)?
  - Bi-directional?

Naively: Let’s Design a Parallel Port

- Requires 12-pins on micro (1 ½ Ports on PSoC)
- Control signals and synchronization?
    - Not bi-directional without control signals (would require another 12-pins! Now we would be completely out!)
    - Receiving PC would have to either poll or service interrupts on any pin change to detect new data

Serial Comm. Interface (UART)

- Need something to convert parallel data to a serial stream
- UART (Universal Asynchronous Receiver/Transmitter)
  - Bi-directional (Receiver/Transmitter)
  - Handles serialization of data (typ. shift registers)
  - Handles data framing (clocks, control signals, etc.)
  - Configurable data format, framing, rate, etc.
- Most modern uCs have built-in UART capabilities
- Signal level compatibility handled by driver

Asynch. Communication

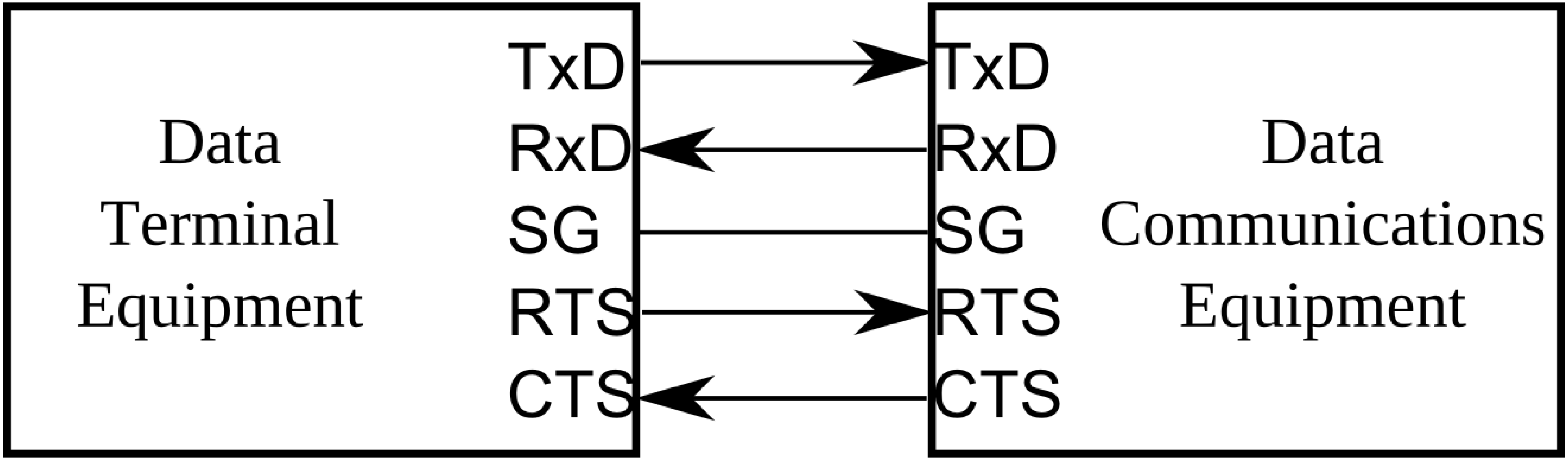
1) Mark & Space: Logic 1 is called mark, logic 0 called space
2) Start bit: Change from 1/0 for one bit’s worth of time (signals to receiver to start shifting in bits)
3) Data: The bits, usually in LSB order (0-1-2-...)
4) Parity: Sometimes used to indicate errors (total number of ones odd, odd parity)
5) Stop bit: Final 0/1 transition (may not always be used)

RS-232 Standard

- Defines:
  - Handshaking signals
  - Direction of signal flow
  - Types of communications devices (DCE & DTE)
  - Connectors and interface mechanical considerations (ie. DB9, DB25)
  - Electrical signal levels
- DCE - Data Communications Equipment (Modems)
- DTE - Data Terminal Equipment (CPUs)
- Data at 20Kbit/s up to 50 ft (why in a minute)

DE9 - Pinout for RS232

1) **DCD**: *Digital Carrier Detect* (Indicates DCE has detected a carrier)
2) **RxD**: *Received Data*
3) **TxD**: *Transmitted Data*
4) **DTR**: *Data Terminal Ready* (Indicates DTE is ready for sending/receiving)
5) **SG**: *Signal Ground*
6) **DSR**: *Data Set Ready* (DCE has made connection and is ready to receive)
7) **RTS**: *Request to Send* (Asserted by DTE to send data)
8) **CTS**: *Clear to Send* (Must be asserted before DTE can transmit)
9) **RI**: *Ring Indicator* (Indicates a ringing signal)



Data Terminal Equipment

TxD

RxD

SG

RTS

CTS

TxD

RxD

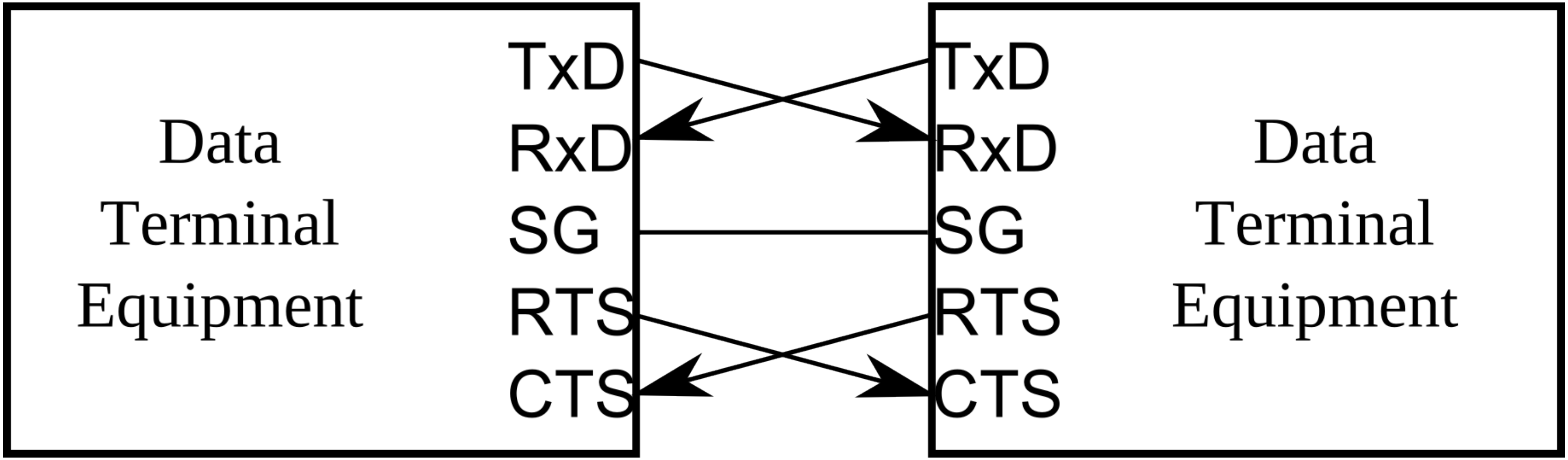
SG

RTS

CTS

Data Communications Equipment

Standard Serial Cable



Data Terminal Equipment

TxD

RxD

SG

RTS

CTS

TxD

RxD

SG

RTS

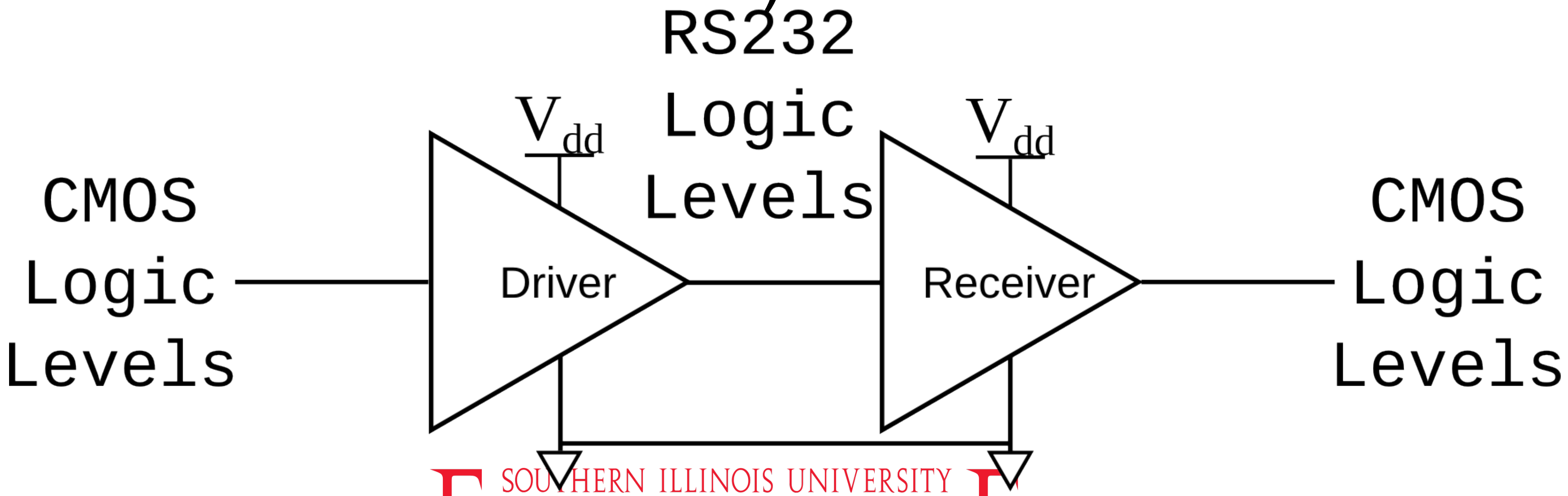
CTS

Data Terminal Equipment

Null Modem

RS232 - Physical Logic Levels

- Mark: -15 to -3V
- Space: +3V to +15V
  - Why so high? (Larger noise margin)
- Spec defines 50 ft, 20 Kbit/s (Amp. slew rate, Common Ground Noise)



RS232 - Configuration

- The UARTs must share the same:
  - Baud Rate (#symbols/sec - essentially same clock)
  - Number of data bits (old school ASCII 7-bits)
  - Are there parity checks? (parity Y/N)
  - What kind of parity? (Odd/Even)
  - Number of stop bits (some require more than 1)
- RS232 can be done with only 3 wires (TxD, RxD, SG)
  - Hardware flow control (CTS/RTS) on/off
    - Handshake: DTE sends CTS, DCE responds with RTS
    - Alt. Handshake: DTE sends CTS to send, DCE sends RTS
  - Software flow control (Use ASCII characters XON/XOFF in band)

USB-to-UART

- Most modern computers don't have RS-232 ports anymore
- USB-to-UART converters take the normal, Tx/Rx data lines and packetize/serialize them for USB
- USB device appears to OS as a serial (COM/tty) port
- KITPROG PSoC5 on our dev kit does this for us
  - FTDI 231X series of ICs very common
  - Arduino uses 2nd ATMEGA on newer versions
- If NOT using KITPROG, USBUART user module in PSoC Creator
  - (ie. for micro-USB connector)
  - OR get an external FTDI 231X chip

Assignment

- Textbook (Cady): 12.1-12.8
- UART User Module Datasheet