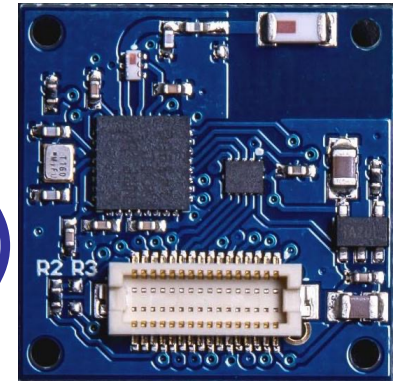
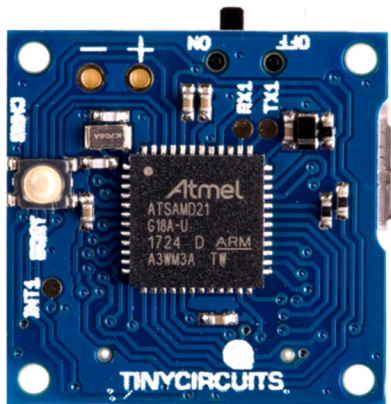


CSE190 Fall 2023

Lecture 10

Serial Busses (cont)



Wireless Embedded Systems

Aaron Schulman

Serial Bus Design Space

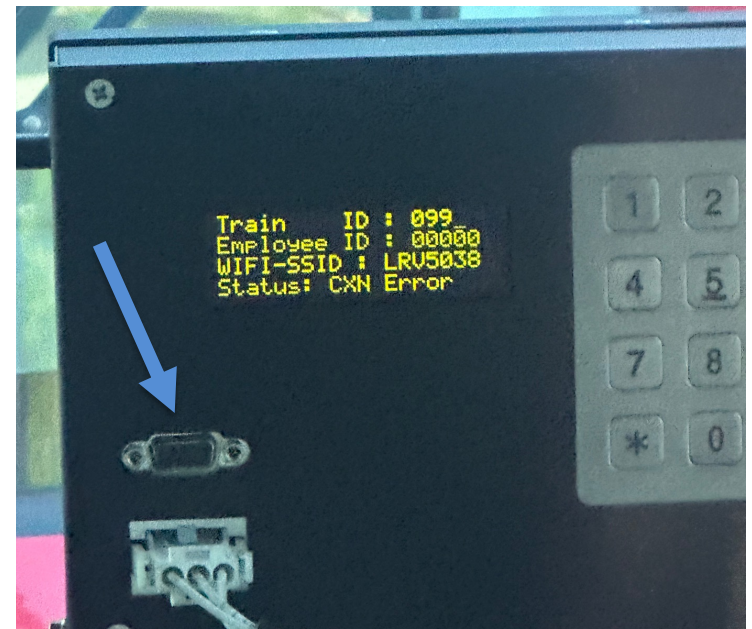
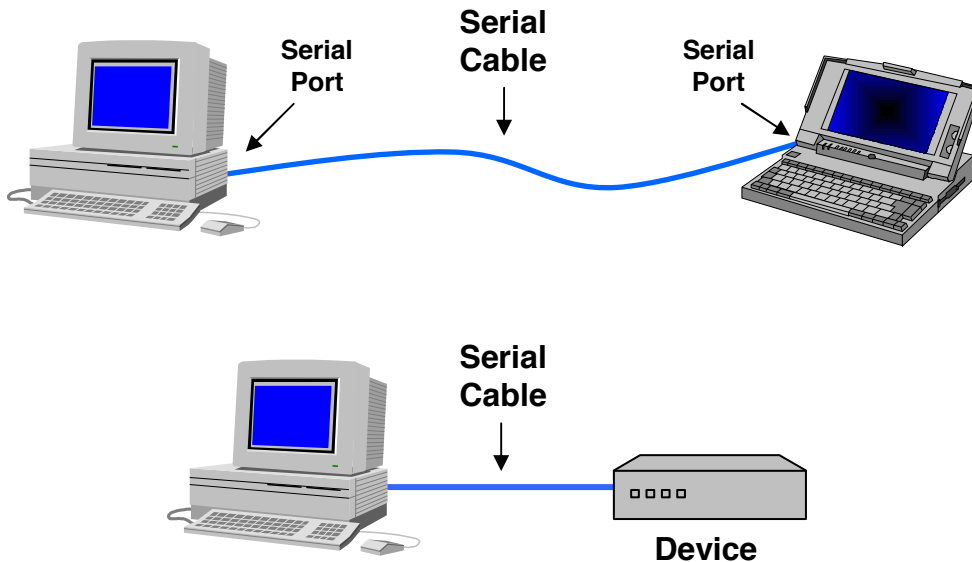
- Number of wires required?
- Asynchronous or synchronous clocking of bits?
- How fast can it transfer data?
- Can it support more than two endpoints?
- Can it support more than one primary?
- How do we support flow control?
- How does it handle errors/noise?
- How far can signals travel?

Serial Bus Examples

	S/A	Type	Duplex	#Devices	Speed (kbps)	Distance (ft)	Wires
RS232	A	Peer	Full	2	20	30	2+
RS422	A	Multi-drop	Half	10	10000	4000	1+
RS485	A	Multi-point	Half	32	10000	4000	2
I2C	S	Multi-primary	Half	?	3400	<10	2
SPI	S	Multi-primary	Full	?	>1000	<10	3+
Microwire	S	Peer	Full	?	>625	<10	3+
1-Wire	A	Peer	half	?	16	1000	1+

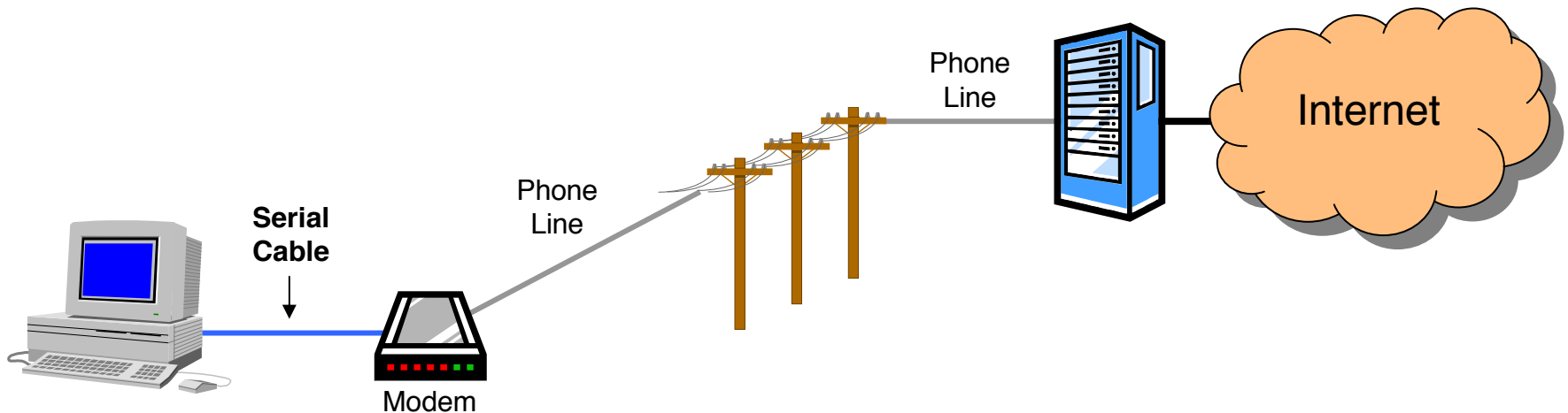
UART Uses

- The PC serial port is a UART!
- Serializes data to be sent over a serial cable
 - De-serializes received data



UART Uses

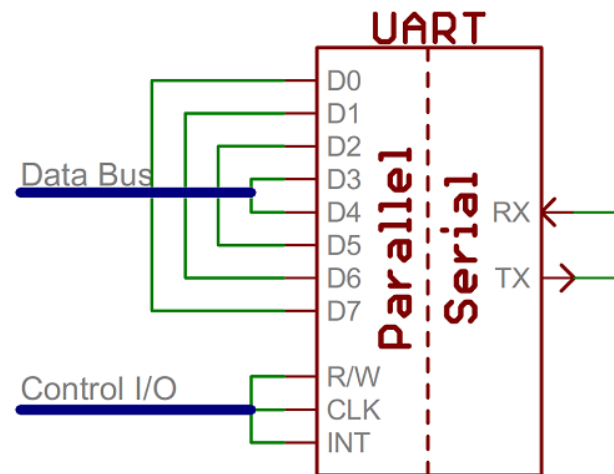
Used to be commonly used for internet access



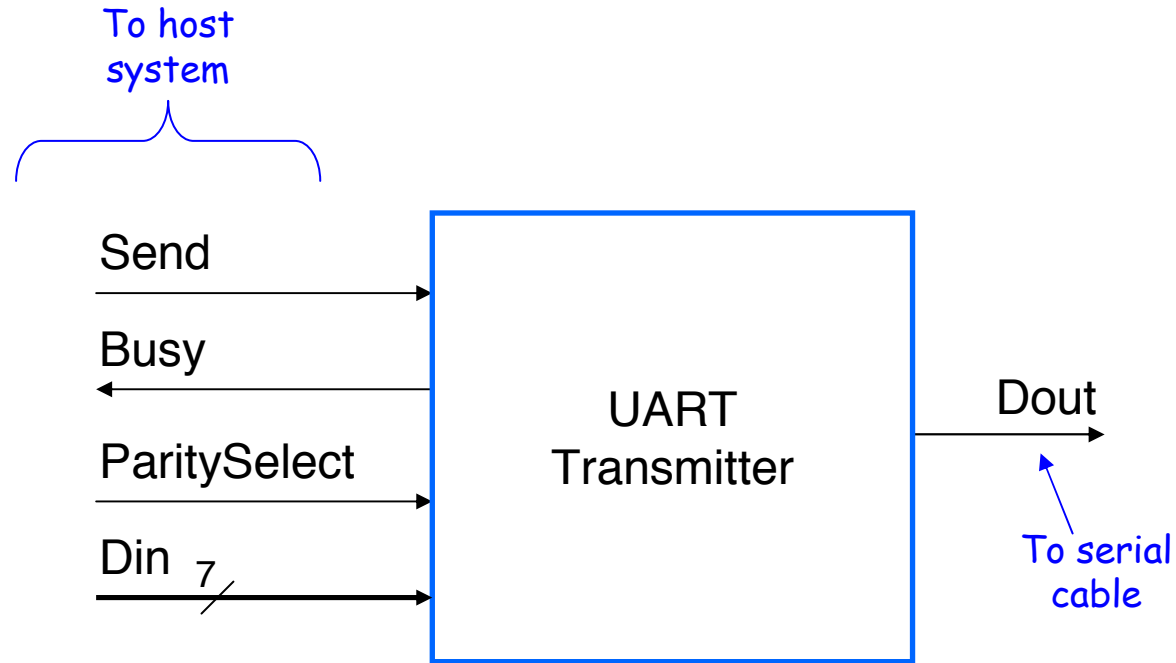
Now often used for **debugging interfaces** on embedded systems (as a debugging terminal)

UART

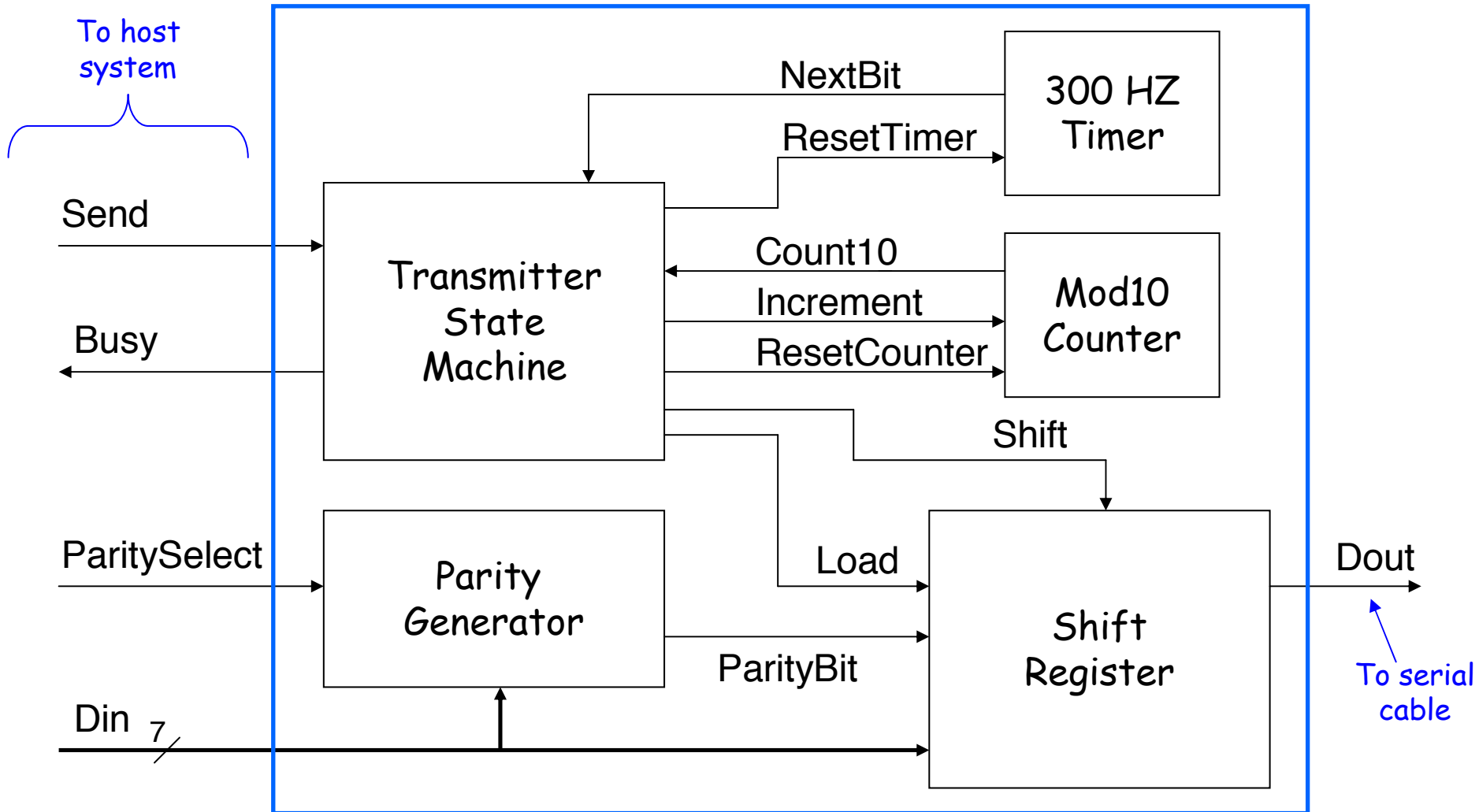
- Universal Asynchronous Receiver/Transmitter
- Hardware that translates between parallel and serial forms
- Commonly used in conjunction with communication standards such as EIA, RS-232, RS-422 or RS-485



Let us design a UART transmitter

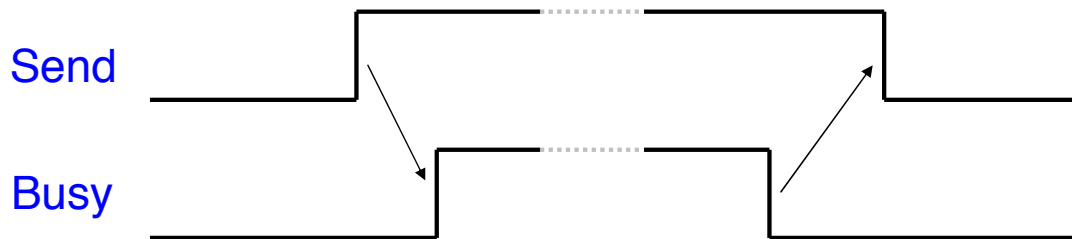


UART Transmitter Block Diagram

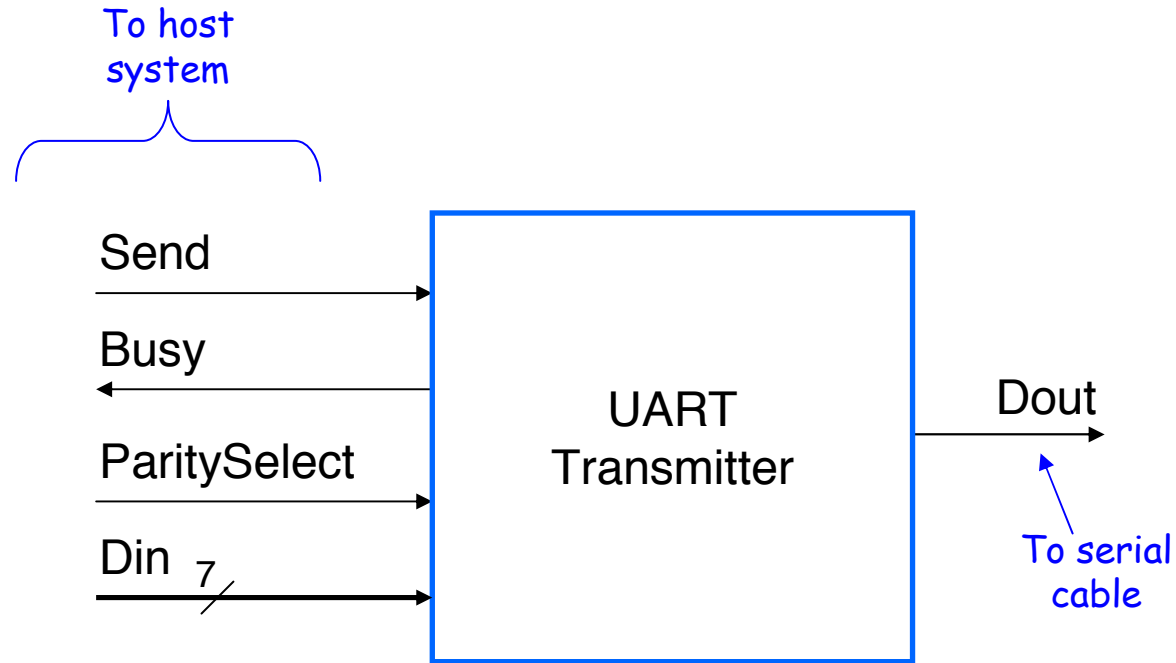


Transmitter/Microcontroller Handshaking

- Microcontroller asserts Send flag and holds it high when it wants to send a byte
- UART asserts Busy flag in response
- When UART has finished transfer, UART de-asserts Busy flag
- (sometimes) system de-asserts Send flag



Let us design a UART transmitter



Transmitter/System Handshaking

- System asserts Send and holds it high when it wants to send a byte
- UART asserts Busy signal in response
- When UART has finished transfer, UART de-asserts Busy signal
- System de-asserts Send signal

