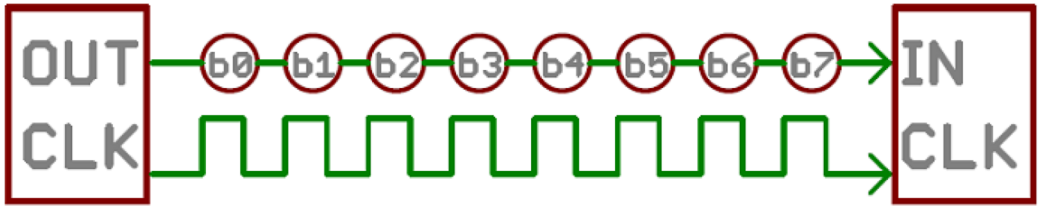
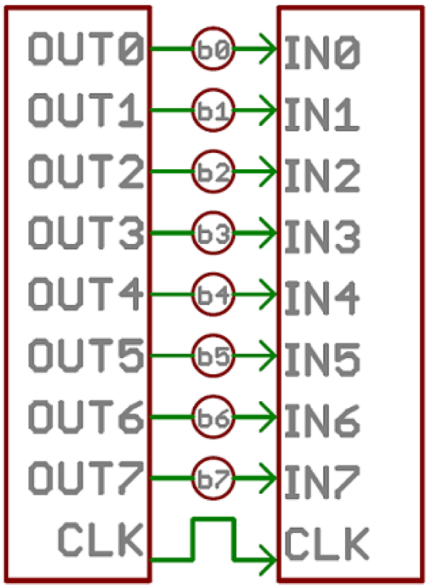
Introduction UART

Communication Protocols

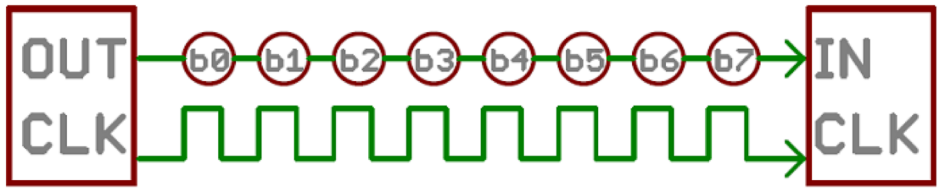
- Parallel Vs. Serial
  - Parallel: Faster / More expensive
  - Serial: Cheaper / slower





Communication Protocols

- Timing in serial communication
  - Synchronous Serial
  - Asynchronous Serial
- Rules of Asynch. Protocols
  - Synch bit
  - Parity bit
  - Baud rate
  - Data bit



Parity and Baud Rate

- Parity
  - Even Parity : 10101010101 → 6 one's → EP=0
  - Odd Parity : 01010101010 → 5 one's → OP=0
- Baud Rate
  - Pulse per second
  - In digital world: Bits per second
  - 9600 baud → 9600 bps or 104 µs per bit.
- Synch bit
  - Required to define the beginning/end of the data
  - the **start bit [LOW]** and the **stop bit [HIGH]**

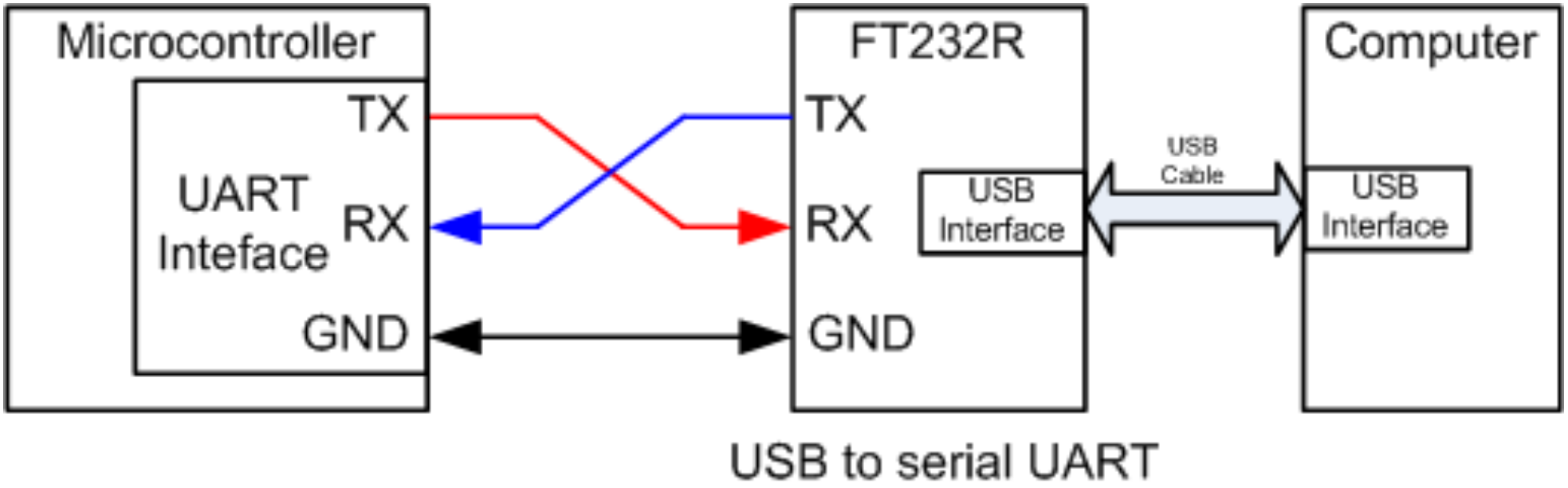
9600 8N1 – 9600 bps/ 8 bits per data / No parity / One Stop bit Always has start bit

Questions

- What will be the EVEN parity value if n=1111 1110?
- What will be the ODD parity value if n=1111 1110?
- How long does it take to transmit a 1-Mbit file at 9600 baud rate?
- Let's say P represents Even Parity. Assume P=1. Can we accept this as a correct frame: 1101 1110P?

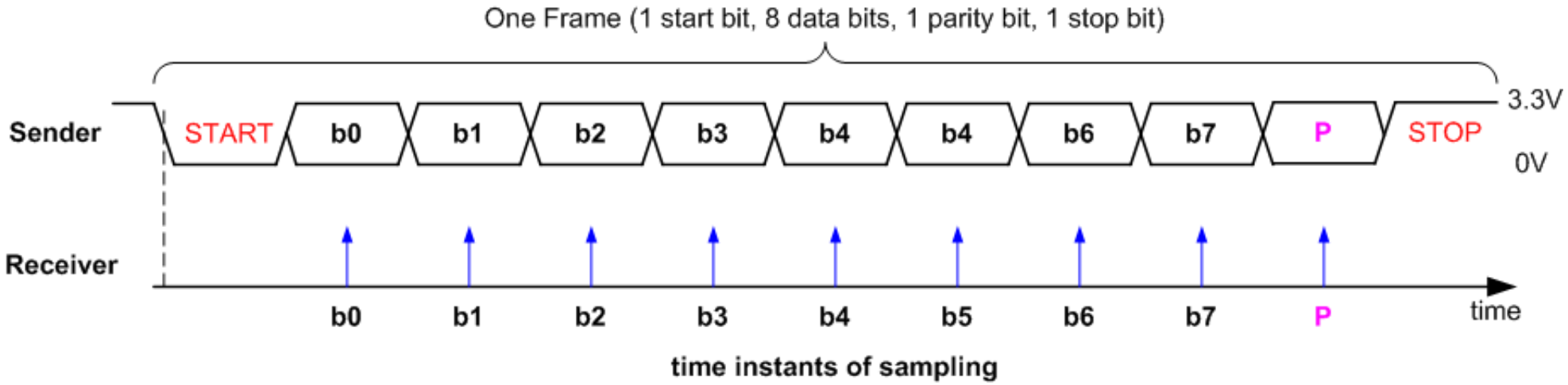
Serial Communication

- Universal Asynchronous Receiver and Transmitter (UART)
  - UART is programmable.
  - Asynchronous
  - Sender provides no clock signal to receivers
  - FT232R converts the UART port to a standard USB interface



USB to serial UART

Data Frame



Tolerate 10% clock shift during transmission

- Sender and receiver uses the same transmission speed
- Data frame
  - One start bit
  - Data (LSB first or MSB, and size of 7, 8, 9 bits)
  - Optional parity bit
  - One or two stop bit

Overhead % =
1 - (Useful Data / Total Data)

9600/(1 + 8 + 1 + 1) = ~872 frames/second

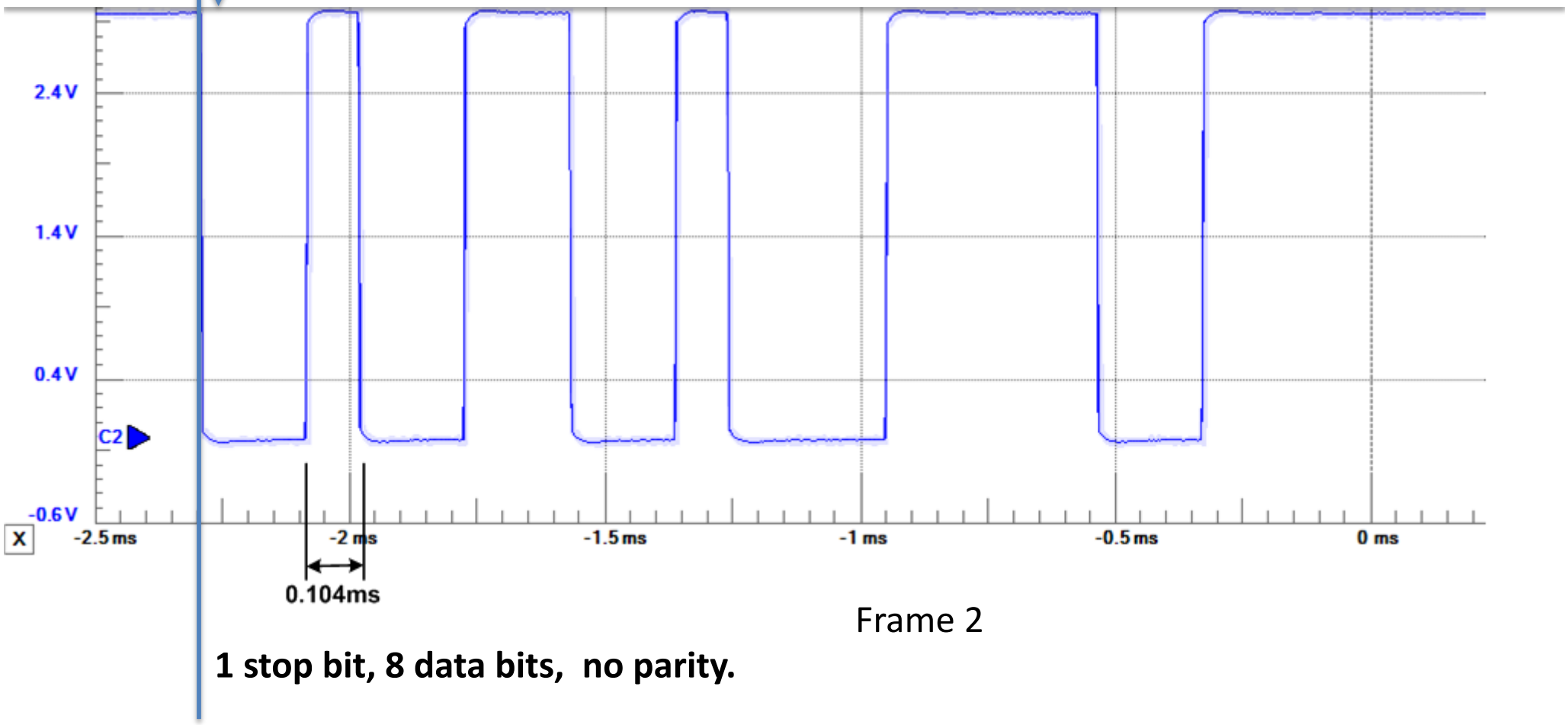
Problem

- Assume we have a file with size 1 million bits. Assume we use 9600:8-1-1 (includes STOP and SATRT bits). How long does it take to transmit the file? Calculate the Overhead %.
  - 8-1-1→ each frame has 11 bits
  - Baud rate is 9600 bits/sec--> Frame rate = 9600/11=872.72 Frames/sec or 1.146 msec/frame
  - Number of frames generated: 1,000,000 bits x 1 frame/8 bit= 125,000 frames
  - Time to transmit 125,000 frames = 125000 frame x 1.146 msec/frame = 143.25 sec!
  - Overhead % = [1 - (Useful Data / Total Data)]x100 = [1 - (8/11)] x 100=27.3

How long does it take to transmit 100,000 bits at 57,600 baud rate?

What are we transmitting? What is the baud rate?

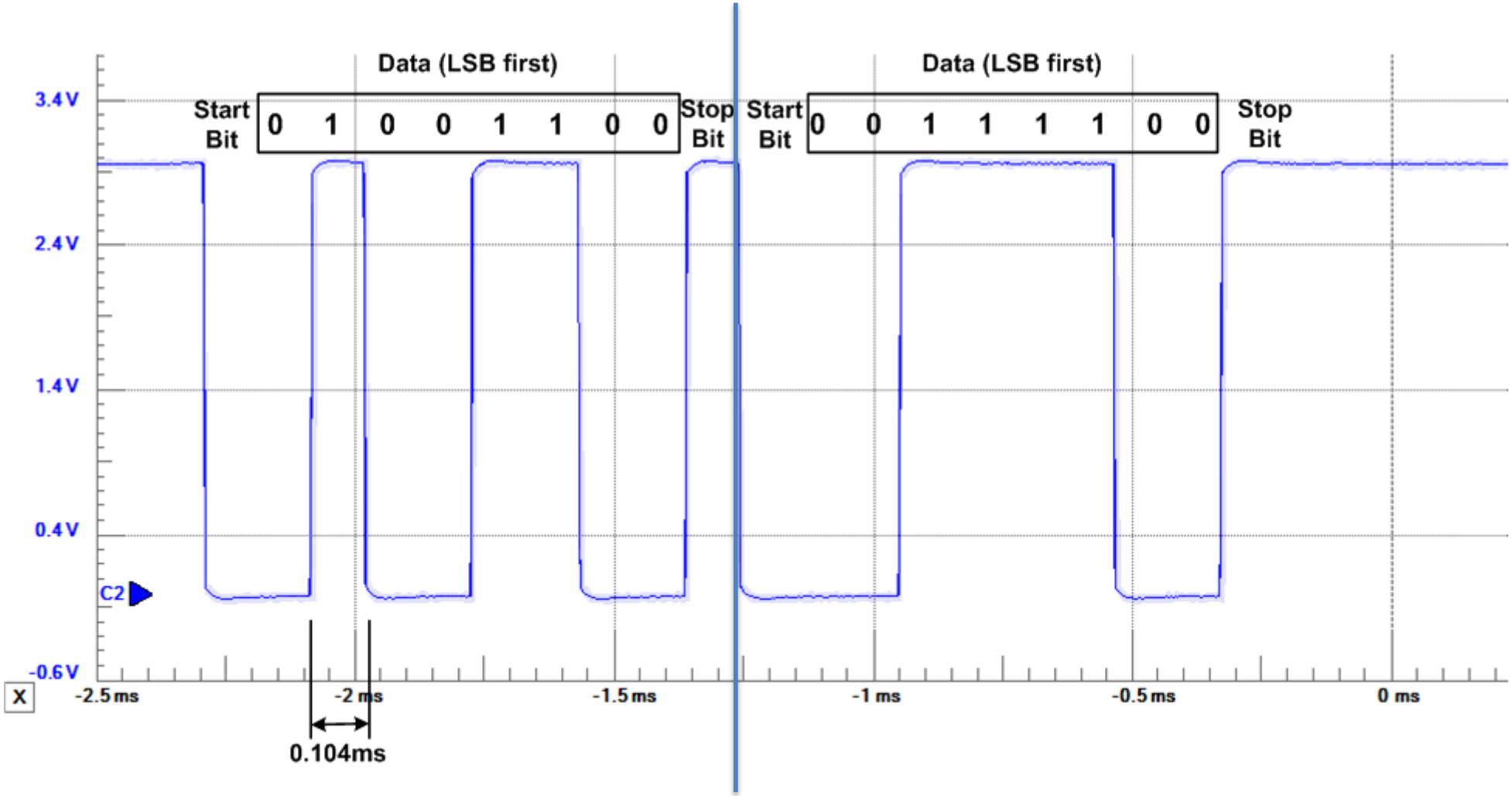
Start here!
Assume LSB first



Frame 2

1 stop bit, 8 data bits, no parity.

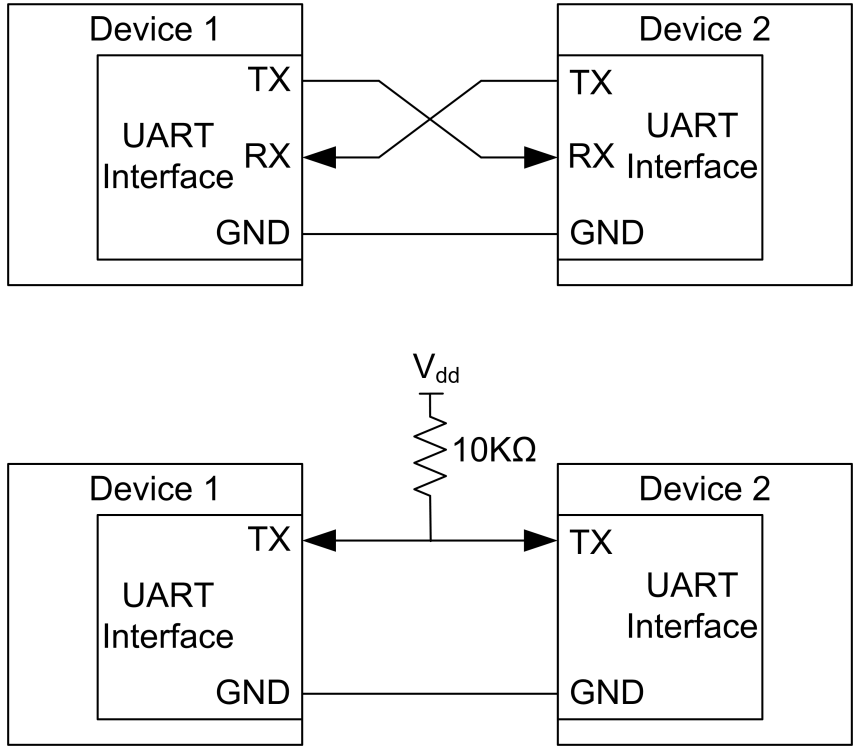
Transmitting 0x32 and 0x3C



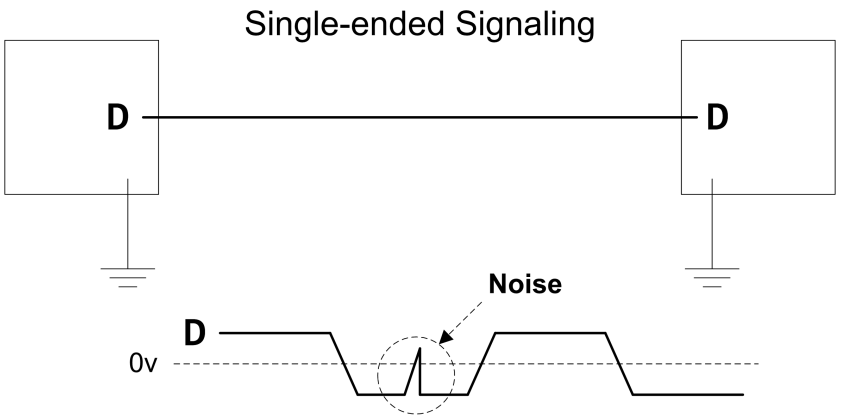
1 start bit, 1 stop bit, 8 data bits, no parity, baud rate = 9600

Communication Modes

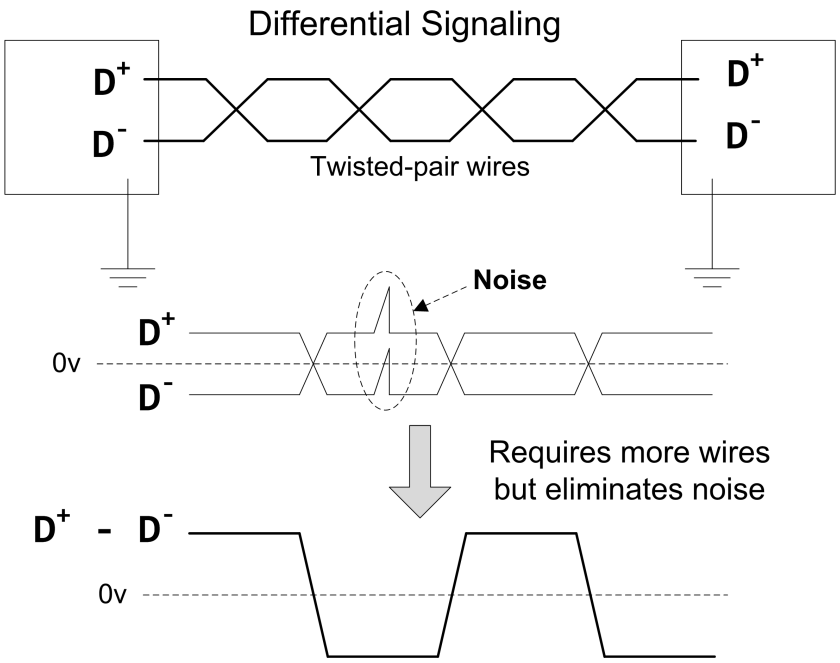
- A serial interface can have different communication modes
  - Full Duplex - means both devices can send and receive simultaneously.
  - Half-Duplex - means serial devices must take turns sending and receiving.

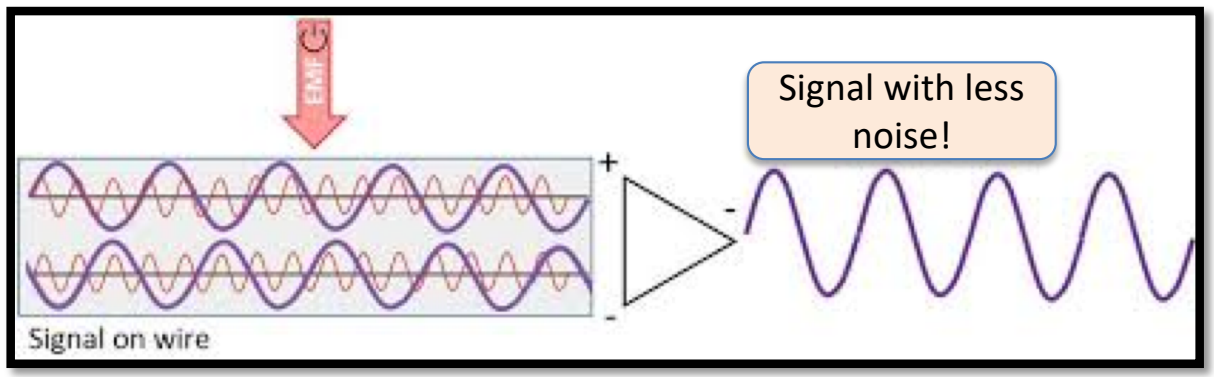


Differential Vs. Single Ended

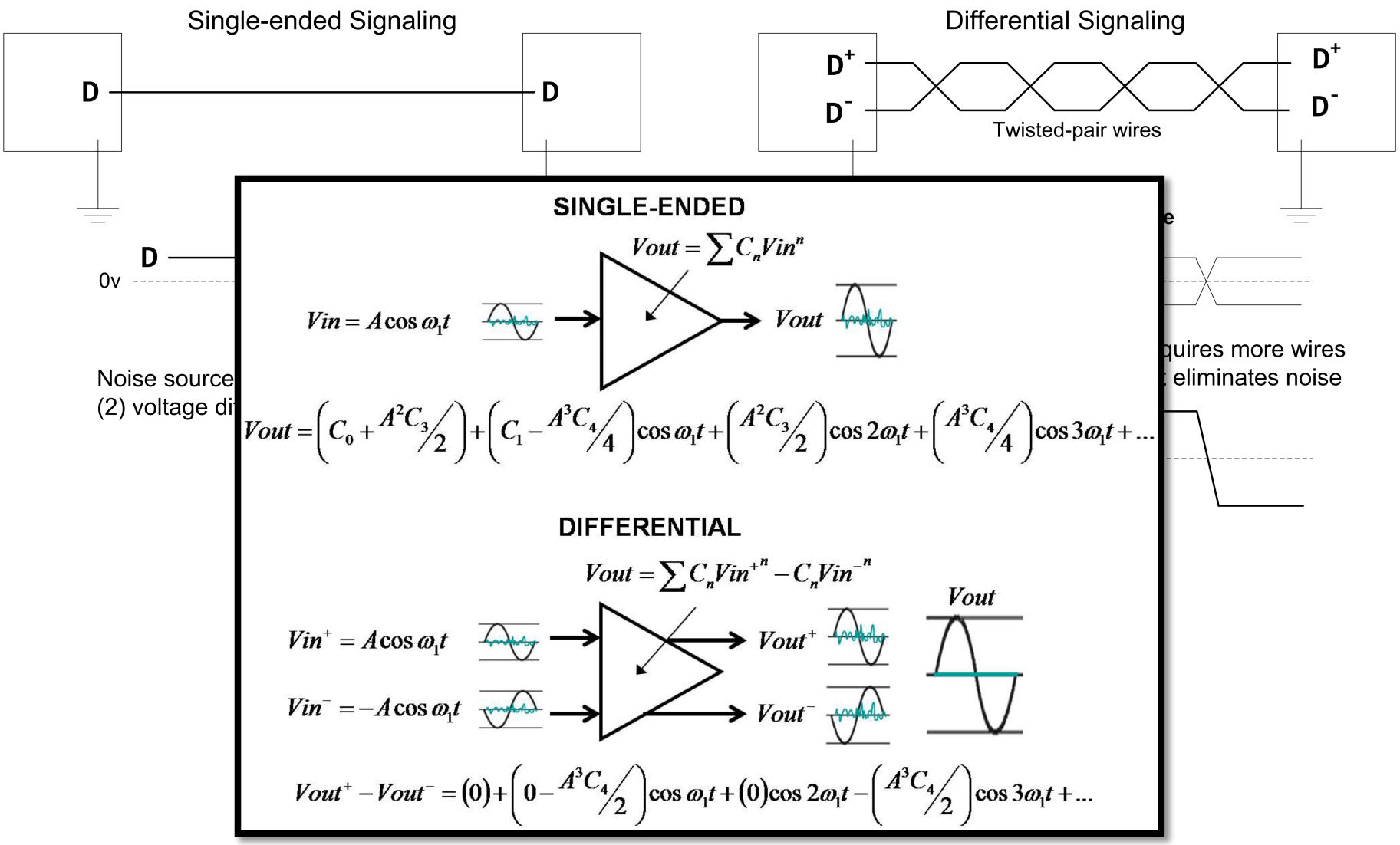


Noise sources: (1) induction picked up on the wired,
(2) voltage difference between two grounds



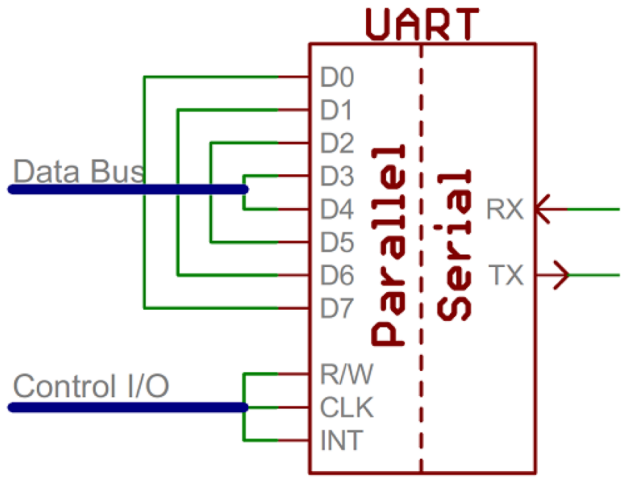


Differential Vs. Single Ended

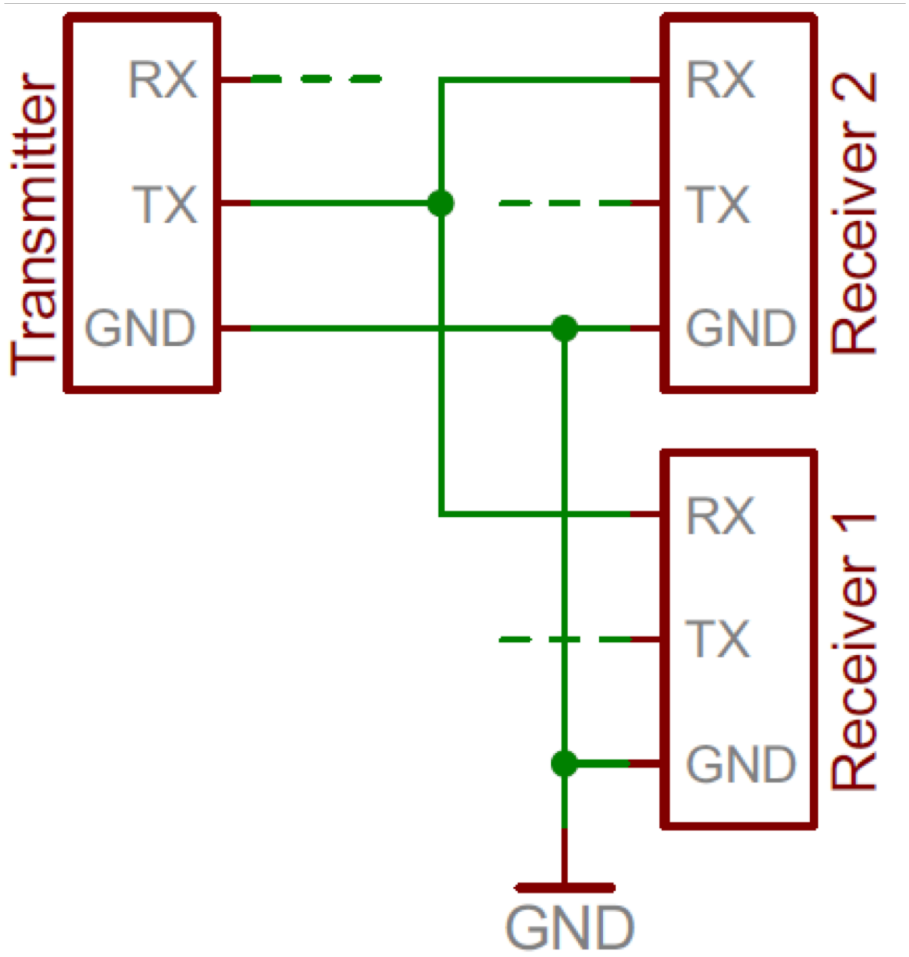


UART

- A universal asynchronous receiver/transmitter (UART) is a block of circuitry responsible for implementing serial communication
- USART – Supports synchronous communication

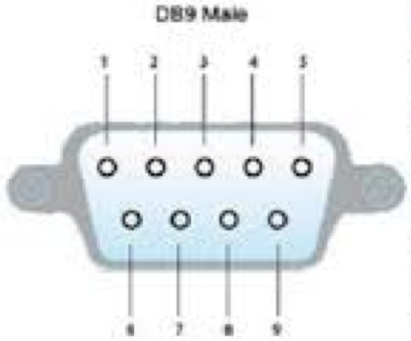


Multiple Bus Connections



UART Protocols
Different Voltage Levels

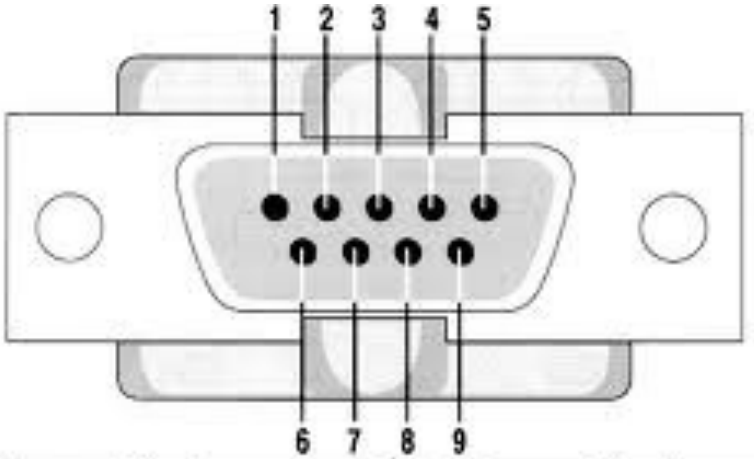
Standard	Voltage signal	Max distance	Max speed	Number of devices supported per port
RS-232	Single end (logic 1: +5 to +15V, logic 0: -5 to -15 V)	100 feet	115Kbit/s	1 master, 1 receiver
RS-422	Differential (-6V to +6V)	4000 feet	10Mbit/s	1 master, 10 receivers
RS-485	Differential (-7V to +12V)	4000 feet	10Mbit/s	32 masters, 32 receivers



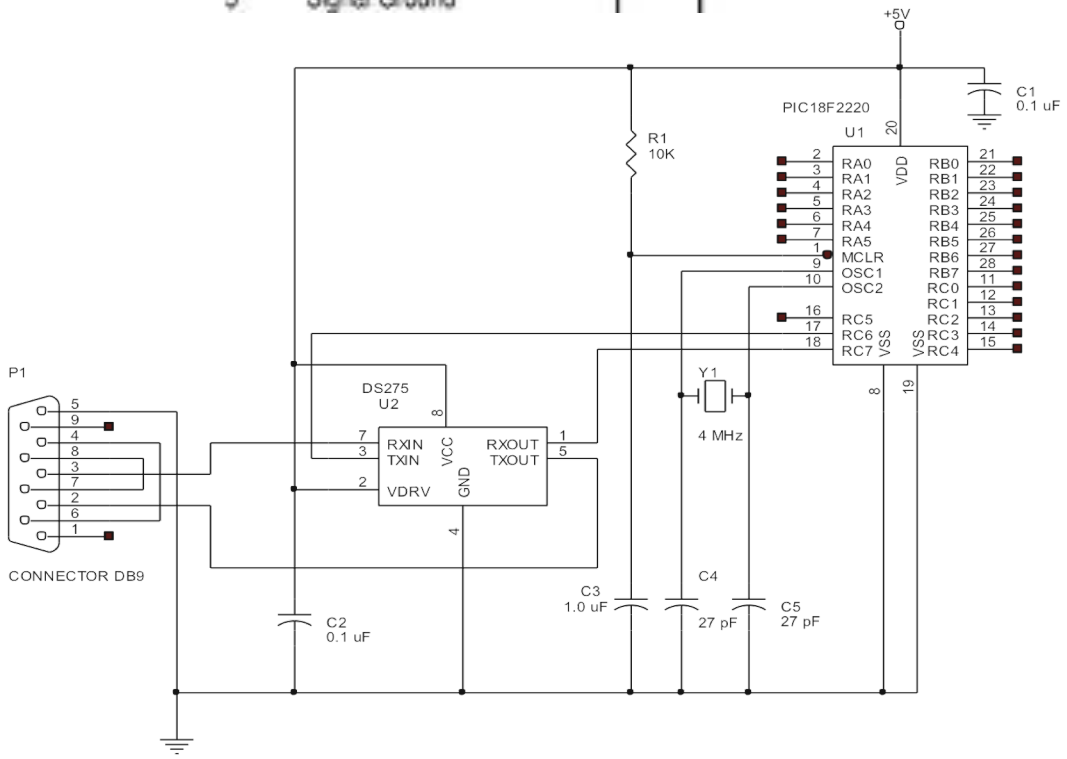
PN	RS-232	RS-485 HALF	RS-485 FULL	RS-422
1	DCD			
2	RXD	DATA-	TXD-	TXD-
3	TXD	DATA+	TXD+	TXD+
4	DTR			
5	GND		GND	GND
6	DSR			
7	RTS		RXD+	RXD+
8	CTS		RXD-	RXD-
9				

Interfacing

- Note that in general we SHOULD use something like Maxim's MAX232 in order to ensure voltage compa)bility between the PIC and the RS232 or the terminal
- It is also possible to INVERT polarity of the signals on TX and RX pins of USART, to interface to the terminal

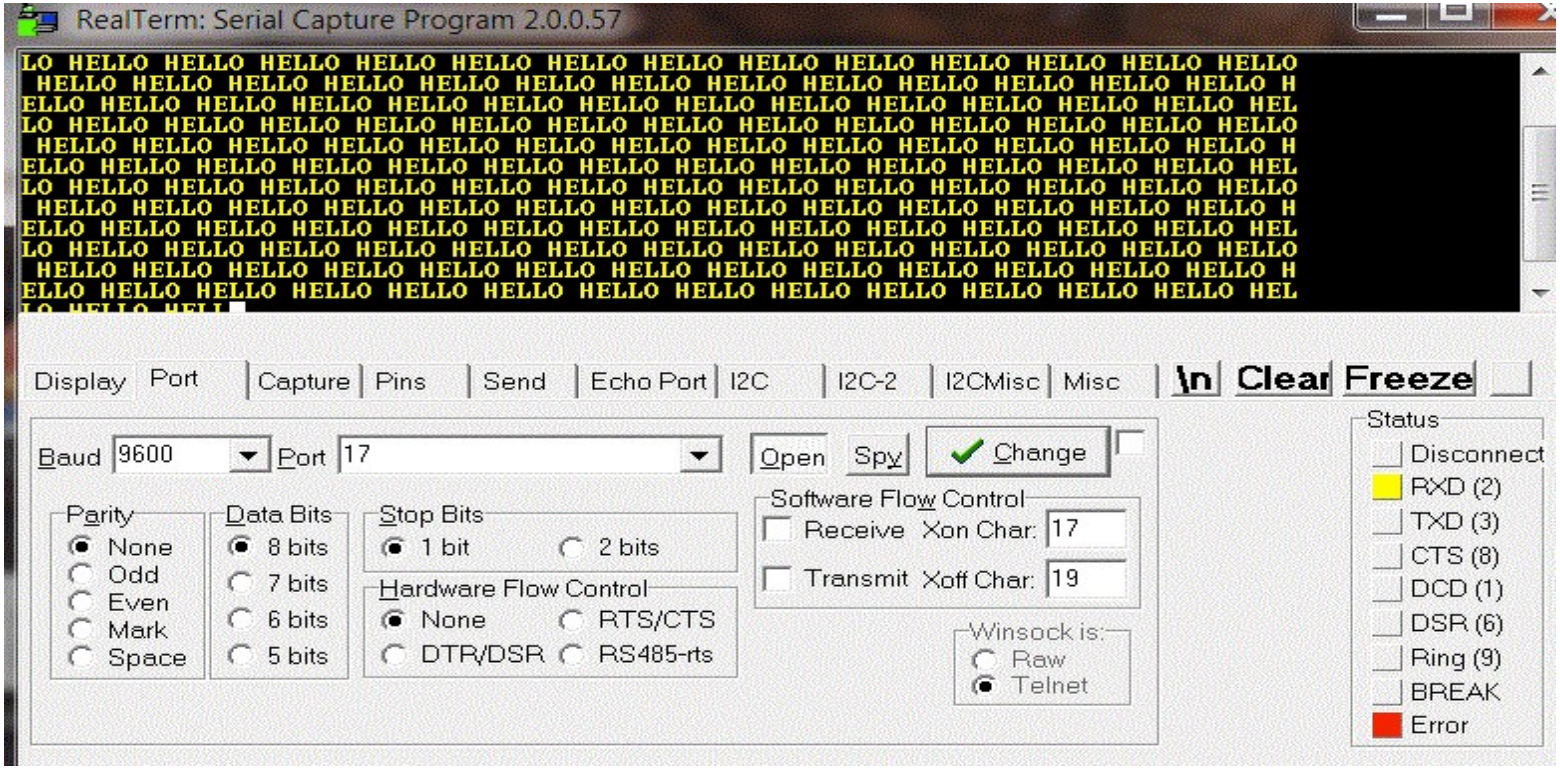


Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		



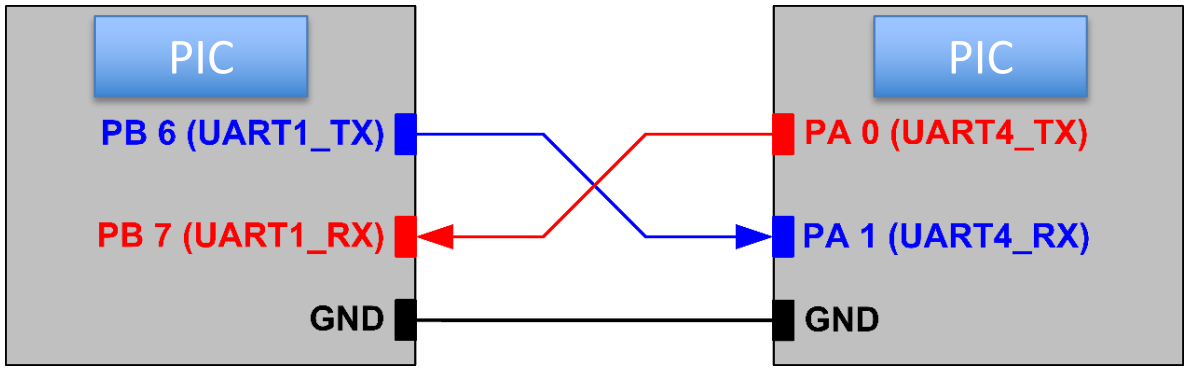
Interfacing to a PC

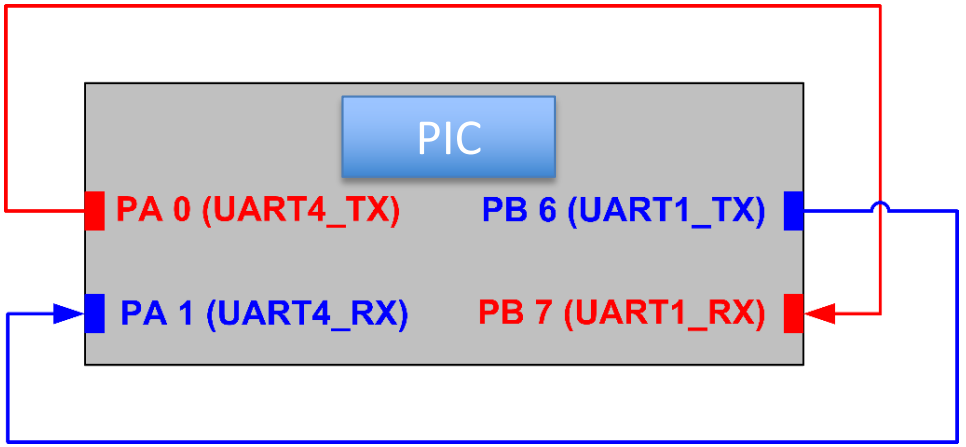
- Download a PC terminal software such as Hyper Terminal or RealTerm (https://sourceforge.net/projects/realterm/) If you only have a USB port you may need a USB/ Serial Cable and driver
- Set the Hyper Terminal to 9600, N,1,0



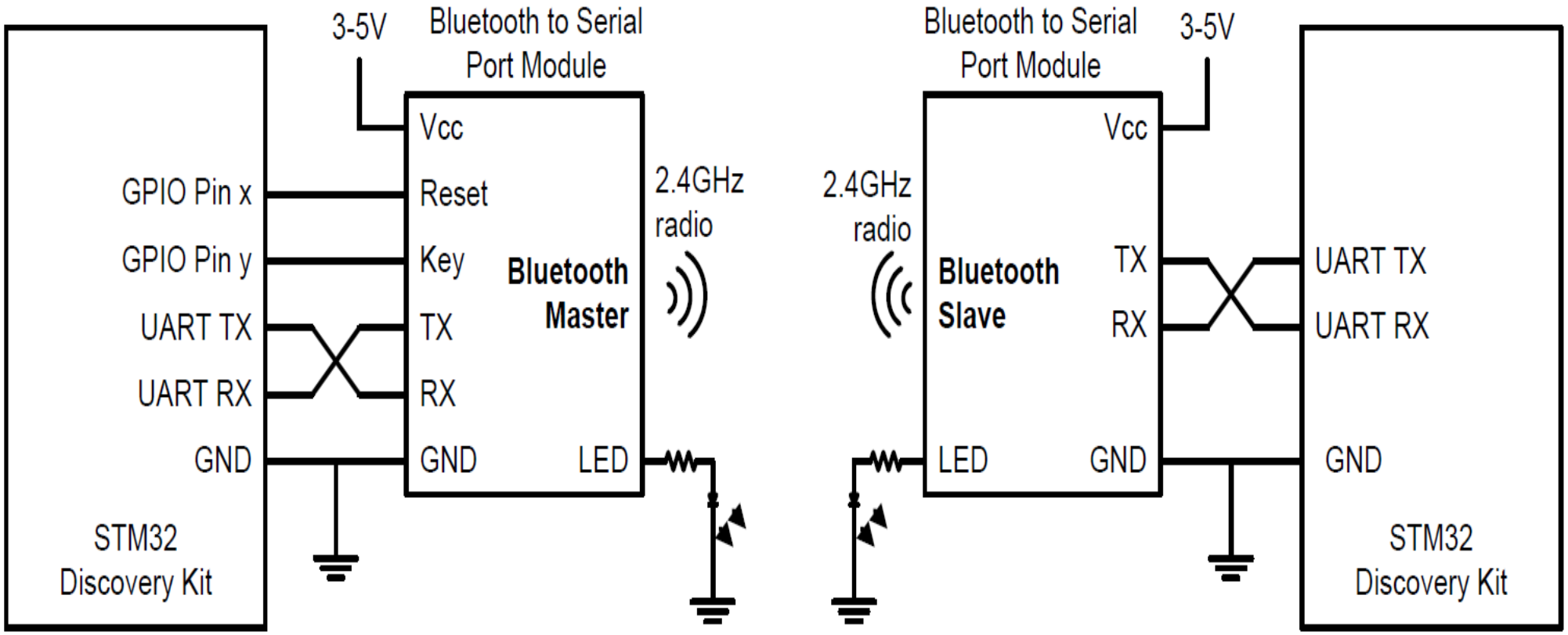
UART Connection

To send and receive data using UART we can use several methods including:
polling, interrupt



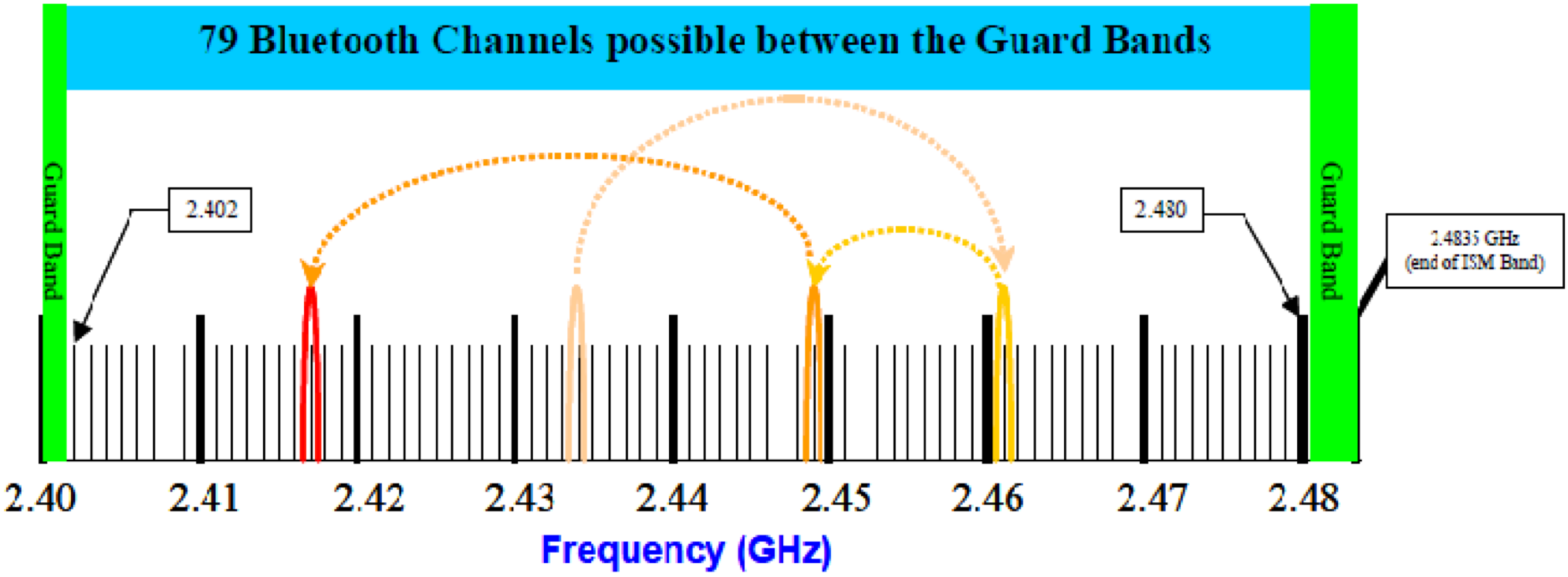


Bluetooth


3-5V
Bluetooth to Serial Port Module
Vcc
Reset
Key
TX
RX
GND
LED
Bluetooth Master
2.4GHz radio
GPIO Pin x
GPIO Pin y
UART TX
UART RX
GND
STM32 Discovery Kit
Bluetooth to Serial Port Module
3-5V
Vcc
2.4GHz radio
Bluetooth Slave
TX
RX
LED
GND
UART TX
UART RX
GND
STM32 Discovery Kit

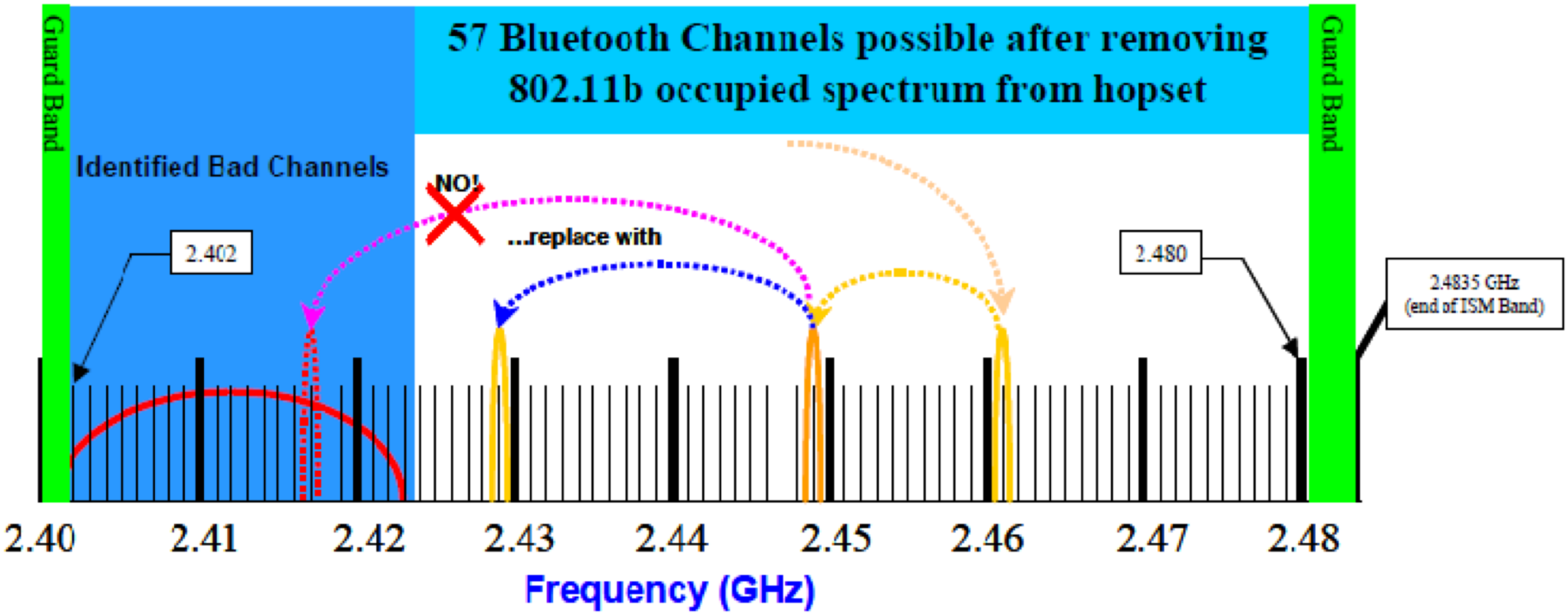
A little about Bluetooth

- Operates at the ISM frequency band
- Bluetooth is divided into 79 channels of different frequencies.
- A Bluetooth device, hops frequency at a rate of 1600 hops per second, randomly selecting a channel of 1 MHz to operate.



A little about Bluetooth

- Operates at the ISM frequency band
- Bluetooth is divided into 79 channels of different frequencies.
- A Bluetooth device, hops frequency at a rate of 1600 hops per second, randomly selecting a channel of 1 MHz to operate.



https://sites.google.com/site/nearcommunications/adaptative-frequency-hopping

Bluetooth PICONET

- Each Bluetooth device has a 48-bit address:

0x 0018 E4 0C68 0A

- **Temporary Network**
- **Up to 8 Active Devices**
- **Master Coordinates the Piconet and Slaves follow the Master**
- **Each Bluetooth Devices may Operate as Either Master or Slave**