TECH-3812 Advanced Electronic Communications
Laboratory Assignment #5 – RS-232
Ver 3.0

Objectives:

- Recognize the difference between TTL RS-232 and standard UART RS-232 signals
- Interpret the transmitted serial signal
- Determine Baud Rate from given signal

Equipment:

- Preprogrammed Arduino (with USB cable to power the Arduino)
- RS-232 9-pin (Male) to 3 pin output (Yellow: Pin 2, Red: Pin 3 and White: Pin 5)
- Jumper Wires
- Rigol Scope
- USB Thumb Drive (Formatted as FAT32)

Background:

RS232 is a standard protocol used for serial communication, it is used for connecting computer and its peripheral devices to allow serial data exchange between them. As EIA defines, the RS232 is used for connecting **Data Transmission Equipment (DTE)** and **Data Communication Equipment (DCE)**.[1]

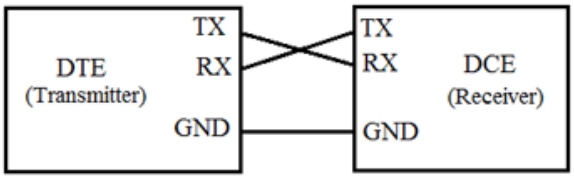


Figure 1 - DTE to DCE Communications

Typically, a Universal Asynchronous Data Receiver &Transmitter (UART**)** is used to convert parallel data into a serial bit stream, at a known transfer (BAUD) rate while also adding start, stop and parity bits, and the receiving UART converts back the serial data back into parallel for the device.

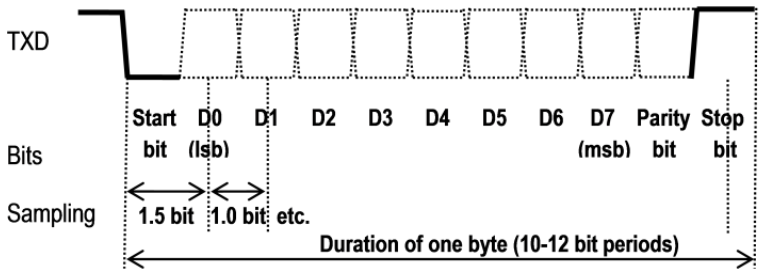


Figure 2 - RS-232 data Transmission

[1] https://circuitdigest.com/article/rs232-serial-communication-protocol-basics-specifications

Parity is an optional bit that is used for error detection. The options are NONE (off), Even or Odd. If using even parity, the bit will be set accordingly to make the number of data bits turned on an even value (counting itself). If using odd parity, the bit will be set accordingly to make the number of data bits turned on be an odd value. Let's take the example of transmitting a "J" (ascii code 74 dec or 0x4A hex) below:

	D0							D7	Parity Bit
Even	0	1	0	1	0	0	1	0	1
Odd									0

Baud rate can be calculated as shown in Figure 3 and shows the relationship between the bit period and baud rate.

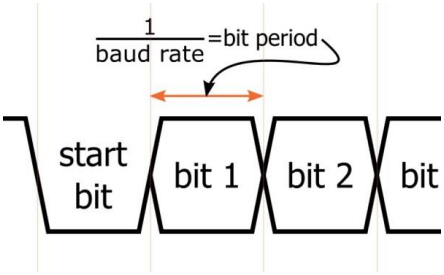


Figure 3 - BAUD Rate

RS-232 standard defines the voltage levels as:

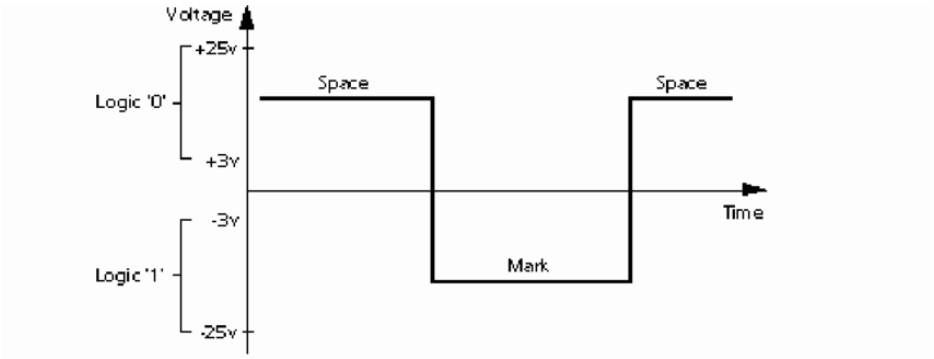


Figure 4- RS-232 Standard Voltage Levels[2]

Note that the RS-232 standard allows quite a large voltage range for Space (Logic 0) and Mark (Logic 1) to allow for signal loss over the cable. Also note that the signal is INVERTED!

But sometimes devices output a TTL Signal (0-5v) and not the voltage as described by the RS-232 Standard. This is called RS-232 TTL and is different. Look at Figure 5:

[2] https://www.researchgate.net/figure/RS-232-Voltage-Levels_fig6_255591153

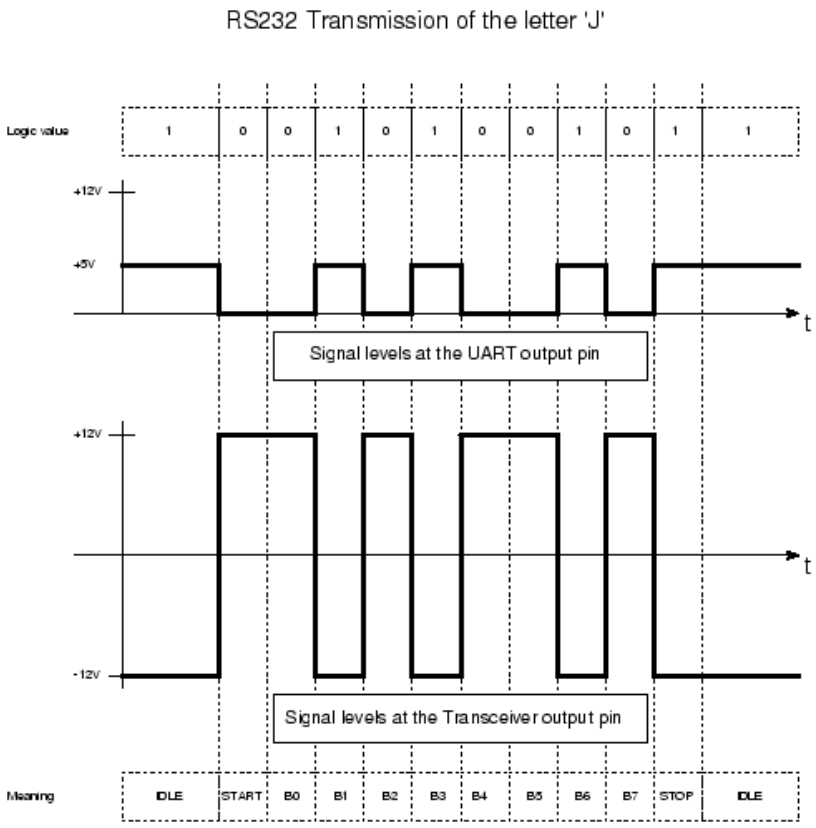


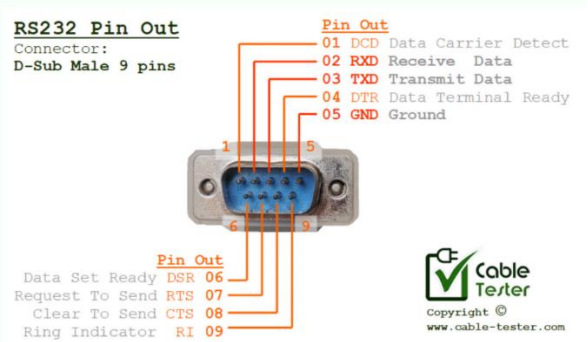
Figure 5 - Comparison of RS232 TTL vs RS232 Signals

Both waveforms show the transmission of the Letter J. The top waveform is the TTL (0-5v) RS-232 Level (with Start being a Low, Data bits being NONE INVERTED (ie high =1) and Stop bit being high).

The bottom signal is the RS-232 Standard signal (showing both positive and negative voltages) with all bits being INVERTED signals (ie high=0).

So looking at an oscilloscope output, given the information above, you should be able to determine the type of signal (TTL or Standard RS-232), Baud Rate, and the message being sent and if parity is used and if so if it is even or odd (assuming no error).

The standard connector is as follows:



TECH-3812 Advanced Electronic Communications
Laboratory Assignment #5 – RS-232
Ver 3.0

Procedure:

NOTE: The Rigol Scope sometimes does not save the trace when doing a screen capture, make sure you check your screen captures before you move on to the next step or you might find you are missing data when you do open the files

1. Connect the following:

Arduino	Scope	Breakout Board	Cable
Pin 1 (Tx)	Ch1	Tx	
	Ch2		Pin 2 (Yellow)
3.3V		Vcc	
Gnd	Gnd	Gnd	

   It is best to use a small protoboard to do multiple connections (Tx and Gnd)
2. Now, connect a jumper wire from Pin 8 to Gnd on the Arduino board and connect the USB cable from the Arduino to a computer in the lab (do **NOT** use the Rigol Scope USB Port!).

If you connected the USB cable before putting in the jumper…press the reset button on the Arduino before continuing.

3. Load "RS232_setup1.stp" setup file (on USB Stick) onto the Rigol Scope.
4. Hit single and capture the waveform.
5. Analyze the captured waveforms by answering the following:
   a. For the two waveforms, What type of Transmission (RS-232 TTL or Standard) is this? How did you determine your answer?
   b. What is the Baud Rate? Show your work (including how you got values from the O-Scope capture)!
   c. What value was transmitted (in binary, hex and ASCII)?
   d. What is the parity setting?
6. Now add a jumper from Pin 9 to grd. Press the Reset Button on the Arduino, then capture the waveform. Analyze it and answering the following:
   a. What is the Baud Rate? Show your work!
7. Remove Pin 9 to ground wire. Now connect Pins 12 and 13 to Ground. Press the Reset Button on the Arduino, then capture and analyze the waveform by answering the following:
   a. What is the parity setting now? How did you determine it?
8. Now remove the ground wires from Pins 8, 12 and 13.
9. Press the Reset Button on the Arduino
10. Load "RS232_setup2.stp" setup file (on USB Stick) onto the Rigol Scope.
11. Find the "Trigger" controls section on the O-Scope. Press the Menu button and select Data and set the value to 85 and hit Single.
12. Capture a screen shot.
13. Repeat setup 9 for values 111, 102 and 77, capturing a screen shot each time.
14. Analyze each of the captures (showing stop, start, 1's and 0's, and data position D0 -> D7) on the capture (using paint).
15. Show the binary value, Hex value and Ascii of each output