In this section we will develop a simple device driver using the Universal Asynchronous Receiver/Transmitter (UART). This serial port allows the microcontroller to communicate with devices such as other computers, printers, input sensors, and LCDs. Serial transmission involves sending one bit at a time, such that the data is spread out over time. The total number of bits transmitted per second is called the **baud rate**. The reciprocal of the baud rate is the **bit time**, which is the time to send one bit. Most microcontrollers have at least one UART. The LM4F120/TM4C123 has 8 UARTs. Before discussing the detailed operation on the TM4C, we will begin with general features common to all devices. Each UART will have a baud rate control register, which we use to select the transmission rate. Each device is capable of creating its own serial clock with a transmission frequency approximately equal to the serial clock in the computer with which it is communicating. A **frame** is the smallest complete unit of serial transmission. Figure 11.6 plots the signal versus time on a serial port, showing a single frame, which includes a **start bit** (which is 0), 8 bits of data (least significant bit first), and a **stop bit** (which is 1). There is always only one start bit, but the Stellaris® UARTs allow us to select the 5 to 8 data bits and 1 or 2 stop bits. The UART can add even, odd, or no parity bit. However, we will employ the typical protocol of 1 start bit, 8 data bits, no parity, and 1 stop bit. This protocol is used for both transmitting and receiving. The information rate, or **bandwidth**, is defined as the amount of data or useful information transmitted per second. From Figure 11.6, we see that 10 bits are sent for every byte of usual data. Therefore, the bandwidth of the serial channel (in bytes/second) is the baud rate (in bits/sec) divided by 10.

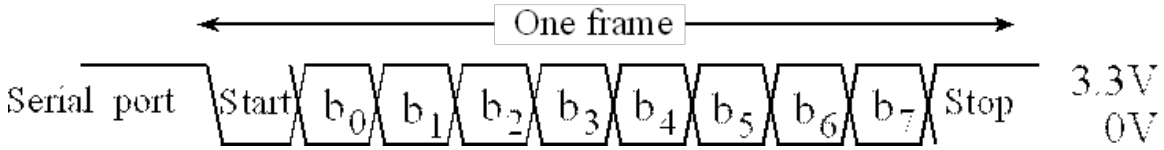

Figure 11.6. A serial data frame with 8-bit data, 1 start bit, 1 stop bit, and no parity bit.

Common Error: If you change the bus clock frequency without changing the baud rate register, the UART will operate at an incorrect baud rate.

CHECKPOINT 11.1

Assuming the protocol drawn in Figure 11.6 and a baud rate of 1000 bits/sec, what is the bandwidth in bytes/sec?

Hide Answer

There is 1 byte of data per 10 bits of transmission. So, there are 100 bytes/sec.

Table 11.1 shows the three most commonly used RS232 signals. The EIA-574 standard uses RS232 voltage levels and a DB9 connector that has only 9 pins. The most commonly used signals of the full RS232 standard are available with the EIA-574 protocols. Only **TxD**, **RxD**, and **SG** are required to implement a simple bidirectional serial channel, thus the other signals are not shown (Figure 11.7). We define the **data terminal equipment** (DTE) as the computer or a terminal and the **data communication equipment** (DCE) as the modem or printer.

DB9 Pin	EIA-574 Name	Signal	Description	True	DTE	DCE
3	103	TxD	Transmit Data	-5.5V	out	in
2	104	RxD	Receive Data	-5.5V	in	out
5	102	SG	Signal Ground			

Table 11.1. The commonly-used signals on the EIA-574 protocol.

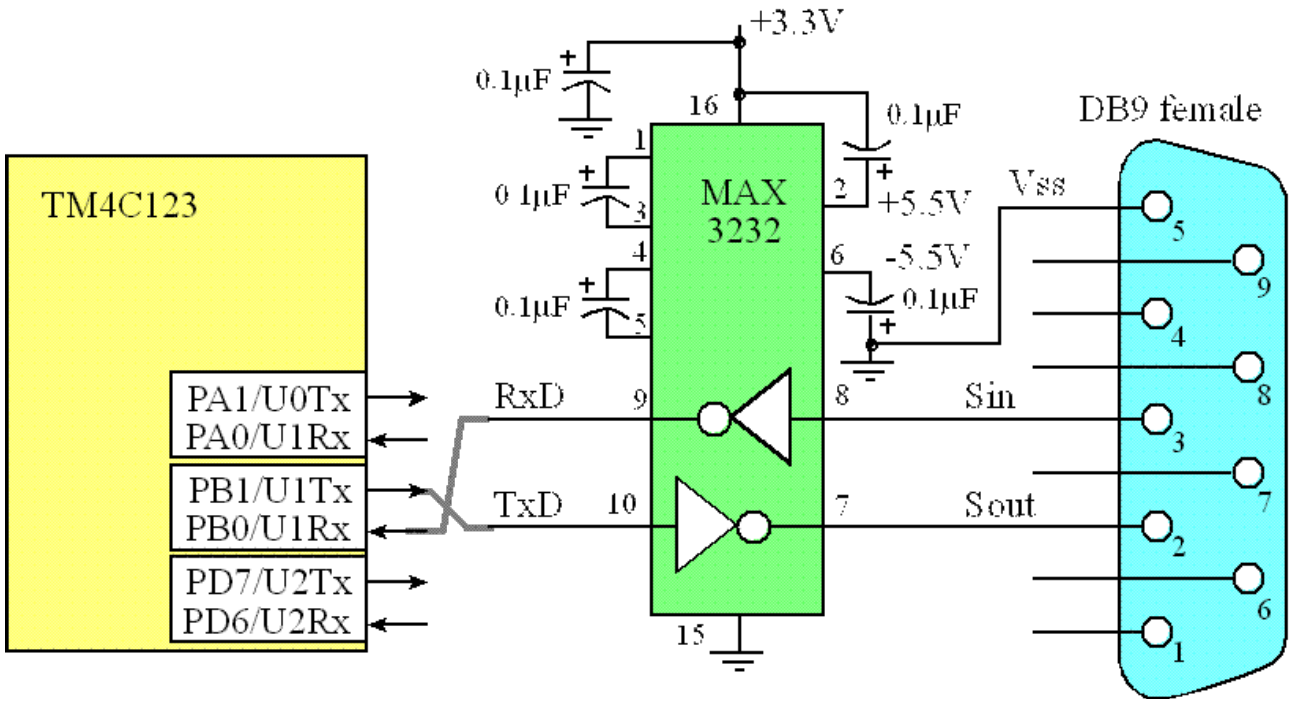

Figure 11.7. Hardware interface implementing an asynchronous RS232 channel. The TM4C123 has eight UART ports.

Observation: The LaunchPad sends UART0 channel through the USB cable, so the circuit shown in Figure 11.7 will not be needed. On the PC side of the cable, the serial channel becomes a virtual COM port. Look in your **Device Manager** to see the port number of your board.

RS232 is a non-return-to-zero (NRZ) protocol with true signified as a voltage between -5 and -15 V. False is signified by a voltage between +5 and +15 V. A MAX3232 converter chip is used to translate between the +5.5/-5.5 V RS232 levels and

the 0/+3.3 V digital levels. The capacitors in this circuit are important, because they form a charge pump used to create the ±5.5 voltages from the +3.3 V supply. The RS232 timing is generated automatically by the UART. During transmission, the MAX3232 translates a digital high on microcontroller side to -5.5V on the RS232/EIA-574 cable, and a digital low is translated to +5.5V. During receiving, the MAX3232 translates negative voltages on RS232/EIA-574 cable to a digital high on the microcontroller side, and a positive voltage is translated to a digital low. The computer is classified as DTE, so its serial output is pin 3 in the EIA-574 cable, and its serial input is pin 2 in the EIA-574 cable. When connecting a DTE to another DTE, we use a cable with pins 2 and 3 crossed. I.e., pin 2 on one DTE is connected to pin 3 on the other DTE and pin 3 on one DTE is connected to pin 2 on the other DTE. When connecting a DTE to a DCE, then the cable passes the signals straight across. In all situations, the grounds are connected together using the SG wire in the cable. This channel is classified as **full-duplex**, because transmission can occur in both directions simultaneously.