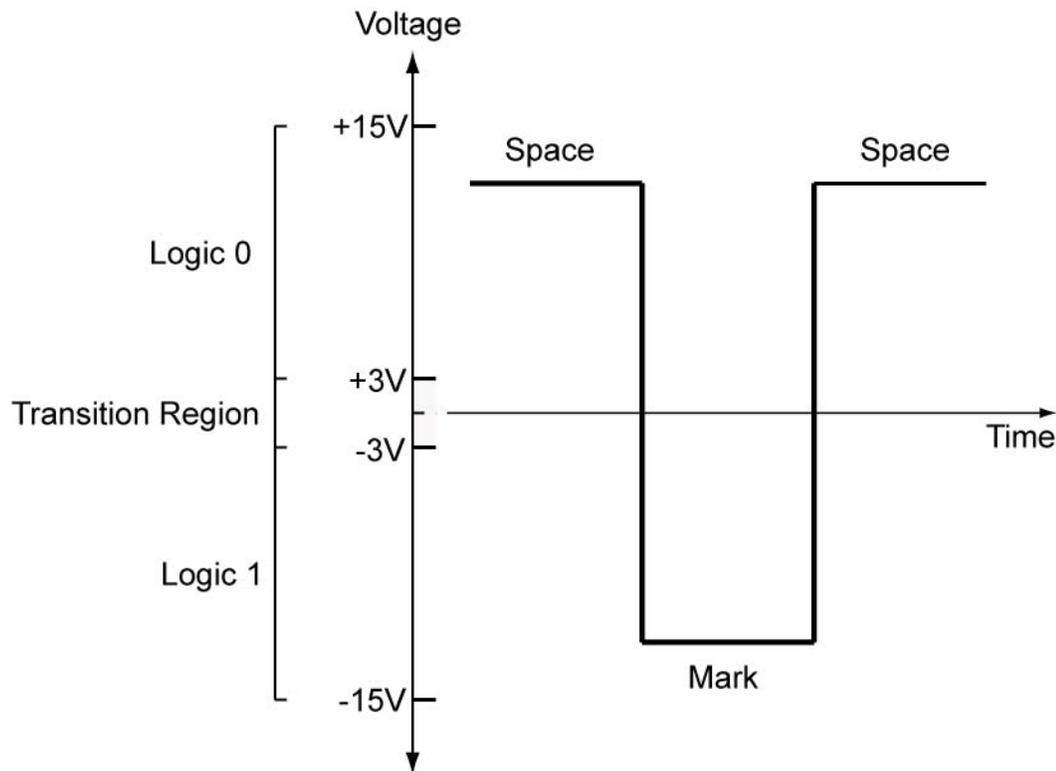


## **RS-232 Serial Communication Tutorial - Hardware Cornerstone Electronics Technology and Robotics II**

- **Administration:**
  - Prayer
- **Related Tutorials:**
  - RS-232 PicBasic Pro Software:  
[http://cornerstonerobotics.org/curriculum/lessons\\_year2/erii\\_rs232\\_2.pdf](http://cornerstonerobotics.org/curriculum/lessons_year2/erii_rs232_2.pdf)
- **RS-232 Serial Communication:**
  - RS-232 (Recommended Standard 232) is a standard for serial binary data signals connecting between a DTE (Data terminal equipment - our PIC microcontroller) and a DCE (Data Circuit-terminating Equipment – our PC).
  - Serial binary communication sends and receives data one bit at a time over one wire. (A parallel communication sends and receives data eight bits at a time over 8 separate wires.)
  - We will use RS-232 serial communication to send and receive data between a PIC microcontroller and a PC.
  - Advantages of the RS-232 as a data interface:
    - RS-232 is a straightforward, universal, and commonly implemented serial interface.
    - Low cost
    - Easy wiring features
  - Disadvantages as a data interface:
    - The standards for RS-232 and similar interfaces usually restrict RS-232 to 20kbps or less.
    - The maximum communication line lengths of 15 meters or 50 feet. The length a data cable depends on speed of the data and the quality of the cable.
  - RS-232 is an asynchronous serial communications protocol used on computers. Asynchronous means it doesn't have any separate synchronizing clock signal, so it has to synchronize itself to the incoming data. It does this by using 'START' and 'STOP' pulses.
  - Besides the synchronization provided by the use of START and STOP bits, an additional bit called a parity bit may optionally be transmitted along with the data. A parity bit affords a small amount of error checking, to help detect data corruption that might occur during transmission. You can choose either even parity, odd parity, mark parity, space parity or none at all. Parity error checking is very rudimentary and we will not use it in this lesson.
  - There are various data types and speeds used for RS-232, the most common type in use, known as 8N1 - the 8 signifies '8 Data Bits', the N signifies 'No Parity' (can also be E 'Even Parity' or O 'Odd Parity'), the final 1 signifies '1 Stop Bit'. The total data sent consists of 1 start bit, 8 data bits, and 1 stop bit - giving a total of 10 bits.
  - The speed we will use is 9600BPS (Bits Per Second), as each byte sent has 10 bits this means we can transfer a maximum of 960 bytes of data per second.

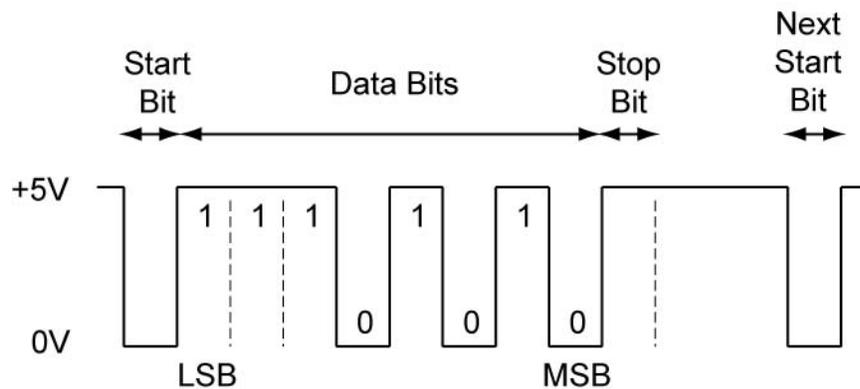
- RS-232 Signal State Voltage Assignments:
  - The RS-232 signal levels are slightly unusual for computers. Rather than the normal 0V to +5V range, it uses +12V to -12V.
  - In the RS-232 standard, voltages of -3v to -15v with respect to signal ground are considered logic '1' (the marking condition). See Figure 26-1 below.
  - Voltages of +3v to +15v are considered logic '0' (the spacing condition).
  - The range of voltages between -3v and +3v is considered a transition region for which a signal state is not assigned.



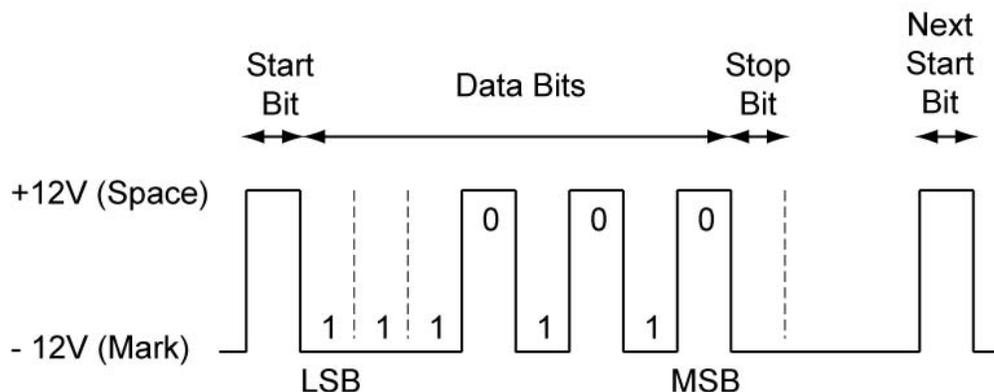
**Figure 26-1: RS-232 Signal State Voltage Assignments**

- The MAX232 integrated circuit is specially designed to interface between 0V to and +5V logic signal levels and the +12V to -12V signal levels of RS-232.

- How to transmit a data byte from the PIC microcontroller to a PC:
  - Initially, the RS-232 signal needs to be at -12V, the 'STOP CONDITION'. The MAX232 inverts a +5V (HIGH or 1) from the PIC to the -12V (or 1) the RS-232 level signal needs. Make sure the PIC output pin is set HIGH in the initialization section of the program. This pin should always be HIGH, **EXCEPT** when we are sending data, when it can be either HIGH or LOW.
  - The RS-232 line is now set at -12V, and the PC is waiting for a 'START BIT'. To generate this all we need to do is set the PIC output pin LOW, the MAX232 inverts the signal and takes the RS-232 line up to +12V. See the Start Bit in Figures 26-2 and 26-3 below.
  - Now we transmit the 8 data bytes, starting with the LSB (Least Significant Bit) bit.
  - That only leaves the 'STOP BIT', for this we set the PIC output pin HIGH. It simply signifies the end of the data byte. If it is the last data byte it could be a considerable time before another 'START BIT' is sent - this is shown in the diagrams by the large gap between the end of the 'STOP BIT' (shown by the dotted line) and the next 'START BIT'.
  - An example of a signal on both sides of a MAX232 converter chip:



**Figure 26-2: Signal as it Leaves the PIC Pin (The PIC Side of the MAX232)**

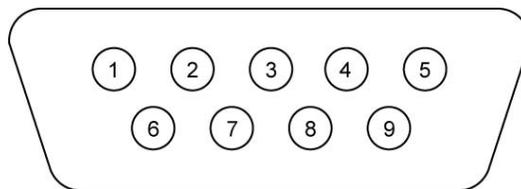


**Figure 26-3: Signal on an RS232 Line (The PC Side of the MAX232)**

- 9-Pin D-Sub Connector:
  - Abbreviations:
    - DTE – Data Terminal Equipment – (the PC)
    - DCE – Data Communications Equipment – (the remote device)
  - 9-Pin D-Sub Connector Pinout:
    - In these RS-232 lessons, we will connect to pins 2,3, and 5 only.



**Figure 26-4: 9 Pin Connector on the PC – Male RS-232 DB9 Viewed from the Front**



**Figure 26-5: 9 Pin Connector Pinout**

Pin Number	Pin Function
Pin 1	DCD - Carrier Detect, from the remote device (DCE) to PC (DTE) DCD indicates that a good carrier is being received from the remote device.
Pin 2	RD - Received Data, from the remote device (DCE) to PC (DTE)
Pin 3	TD - Transmit Data, from PC (DTE) to the remote device (DCE)
Pin 4	DTR - Data Terminal Ready, from PC (DTE) to the remote device (DCE) DTR indicates to the remote device that the PC is on.
Pin 5	SG - Signal Ground reference voltage
Pin 6	DSR - Data Set Ready, from the remote device (DCE) to PC (DTE) DSR is an indication to the PC that the remote device is on.
Pin 7	RTS - Request to Send, from PC (DTE) to the remote device (DCE) RTS is on constantly in most asynchronous situations.
Pin 8	CTS - Clear to Send, from the remote device (DCE) to PC (DTE) RTS is on constantly in most asynchronous situations.
Pin 9	RI - Ring Indicator ringing tone detected from the remote device

**Table 26-1: Pin Function of a RS-232 DB9 Connector**

- A straight-through cable is used to connect a DTE (e.g. computer) to a DCE (e.g. modem), all signals in one side connected to the corresponding signals in the other side in a one-to-one basis. See the table below.

PC (DTE – Data Terminal Equipment)			Remote Device (DCE – Data Communication Equipment)	
Pin Number	Pin Function		Pin Function	Pin Number
1	DCD		DCD	1
2	RD		RD	2
3	TD		TD	3
4	DTR		DTR	4
5	SG		SG	5
6	DSR		DSR	6
7	RTS		RTS	7
8	CTS		CTS	8
9	RI		RI	9

**Table 26-2: Pin Function of a RS-232 Straight-Through Cable**

- A crossover (null-modem) cable is used to connect two DTE directly, without a modem in between. They cross transmit and receive data signals between the two sides and there are many variations on how the other control signals are wired.
- The RS-232 interface presupposes a common ground between the DTE and DCE.
- **RS-232 Level Converters, the MAX232 Chip:**
  - MAX232 is a 0V to +5V to RS-232 converter chip, this converts the 0 to +5V TTL levels at the PIC pins to the +12V to -12V RS-232 signal levels.
  - The MAX232 includes a voltage doubler which generates +10V and a voltage inverter which generates -10V from a single +5V power supply.
  - It also includes two receivers and two transmitters in the same package.
  - Data Sheet: <http://www.maxim-ic.com.cn/pdfserv/en/ds/MAX220-MAX249.pdf>
  - V+ (+10V) and V- (-10V) are not regulated, so the output voltage drops with increasing load current.
  - If you're not sure what Pin does what on the end of your COM lead - check it first before wiring. You could damage your motherboard or MAX232 chip. Grab a multimeter and throw it on resistance, now you can be sure you have the right pin on the other end. See: <http://www.digital-diy.net/18F%20Examples/MAX232.aspx>
  - MAX232 can be used for a negative voltage supply for up to 10 mA. Stack multiple MAX232 IC's in parallel for increased current capabilities.

**Cornerstone Electronics Technology and Robotics II**  
**RS-232 Serial Communication Tutorial – Hardware – Lab 1, MAX232 Converter Chip**

- **Purpose:** The purpose of this lab is to demonstrate the conversion of RS-232 signals to CMOS or TTL signals using the MAX232 converter chip.
  
- **Apparatus and Materials:**
  - 1 – Breadboard with +5V Power Supply
  - 1 – Function Generator
  - 1 - Oscilloscope
  - 1 – MAX232 RS232 Driver/Receiver
    - Different types of MAX232s require different external capacitors. For example, the MAX232A uses 0.1 uF capacitors.
    - MAX233 and MAX233A do not use external capacitors
    - The 10K pull-up resistor R1 helps with the MAX232 noise sensitivity
  - 5 – 1uF Capacitors
  - 1 – 10K Resistor
  
- **Procedure:**
  - Wire the circuit below. **Be very careful** that you install the capacitors with the polarity placed in the correct manner.
  - **RS-232 to TTL:**
    - Connect the oscilloscope Channel 1 to Point D, (RS-232 input), and the common ground.
      - View only Channel 1 trace.
      - Set the Channel 1 VOLT/DIV to 5V.
      - Set the TIME/DIV to 0.2 ms.
      - Position the trace in the center of the screen.
    - Also connect the function generator to Point D, (RS-232 input), and the common ground.
      - Set the waveform to generate a square wave.
      - Set the frequency to approximately 1 kHz.
      - Viewing Channel 1 on the oscilloscope, adjust the amplitude of the square wave such that it is from +10V to -10V. This square wave signal will simulate a RS-232 signal input into the MAX232.
      - Now position Channel 1 trace to top of the oscilloscope screen.
    - Connect the oscilloscope Channel 2 to Point B, (To CMOS or TTL), and the common ground.
      - View both Channel 1 and Channel 2 traces.
      - Set the Channel 2 VOLT/DIV to 5V.
      - Position the trace on the lower half of the screen.
    - Apply power to the circuit and verify that the MAX232 converts the RS-232 signal (+10V to -10V) in Channel 1 to a CMOS or TTL signal (0V to +5V) in Channel 2. Notice the inversion of the signal.

