# EXPERIMENT #12
# RS232 INTERFACE

## INTRODUCTION:

The RS232 interface is one of the most popular communications accessories in electronic equipment. Modems and terminals are not the only devices equipped with this interface; many industrial devices, such as meters, scopes, telemetry systems, and even robots are often equipped with an RS232 interface. All technicians need a working knowledge of the RS232 standard!

This experiment will demonstrate how an RS232 interface works. A full-duplex connection will be established between a UART circuit and a personal computer (PC). When the circuit is working correctly, data transmitted by the UART will be displayed on the screen of the PC. Any keys pressed on the PC will be transmitted down the RS232 back to the UART, which will decode and display the ASCII value for the key pressed.

## CIRCUIT ANALYSIS:

Figure 1 shows the circuitry needed to make the UART "talk" with the PC. The design is very simple, as you can see. Two IC chips, the Motorola MC1488 and MC1489, do all the work.

The UART transmit data comes from "SO," the serial output. This is, of course, TTL-level data. U1, the MC1488, translates the TTL voltages into RS232 voltages (remember, a mark or logic "1" is -3 to -15 Volts nominal, and a space or logic "0" is represented by +3 to +15 Volts nominal.) This circuit operates from +12 and -12 Volt supplies, so naturally, the MC1488 outputs will be close to +12 Volts for a space (0) and -12 Volts for a mark. The translated transmit signal is sent to the PC on pin 3 of the RS232 cable, which is the receive data pin. Software in the PC will read the data and display the equivalent ASCII character on the screen.

When a key is pressed on the IBM, the software in the PC transmits the ASCII equivalent of the character down pin 2 of the RS232 interface, the transmit data pin. U2, the MC1489, converts the RS232 levels back into TTL, and sends the TTL data to the "SI" pin on the UART (which decodes the data and outputs a parallel character to the received data LEDs.)

The IBM-PC must run a special program called a *terminal emulator* during this experiment. The terminal emulator program causes all data received from the IBM's serial port to be sent to the screen, and automatically transmits any keys struck to the transmit data pin of the serial port. The terminal emulator program needed for this experiment is called "EZ-TERM." *EZ-TERM* is supplied on the instructor resource CD for the textbook.

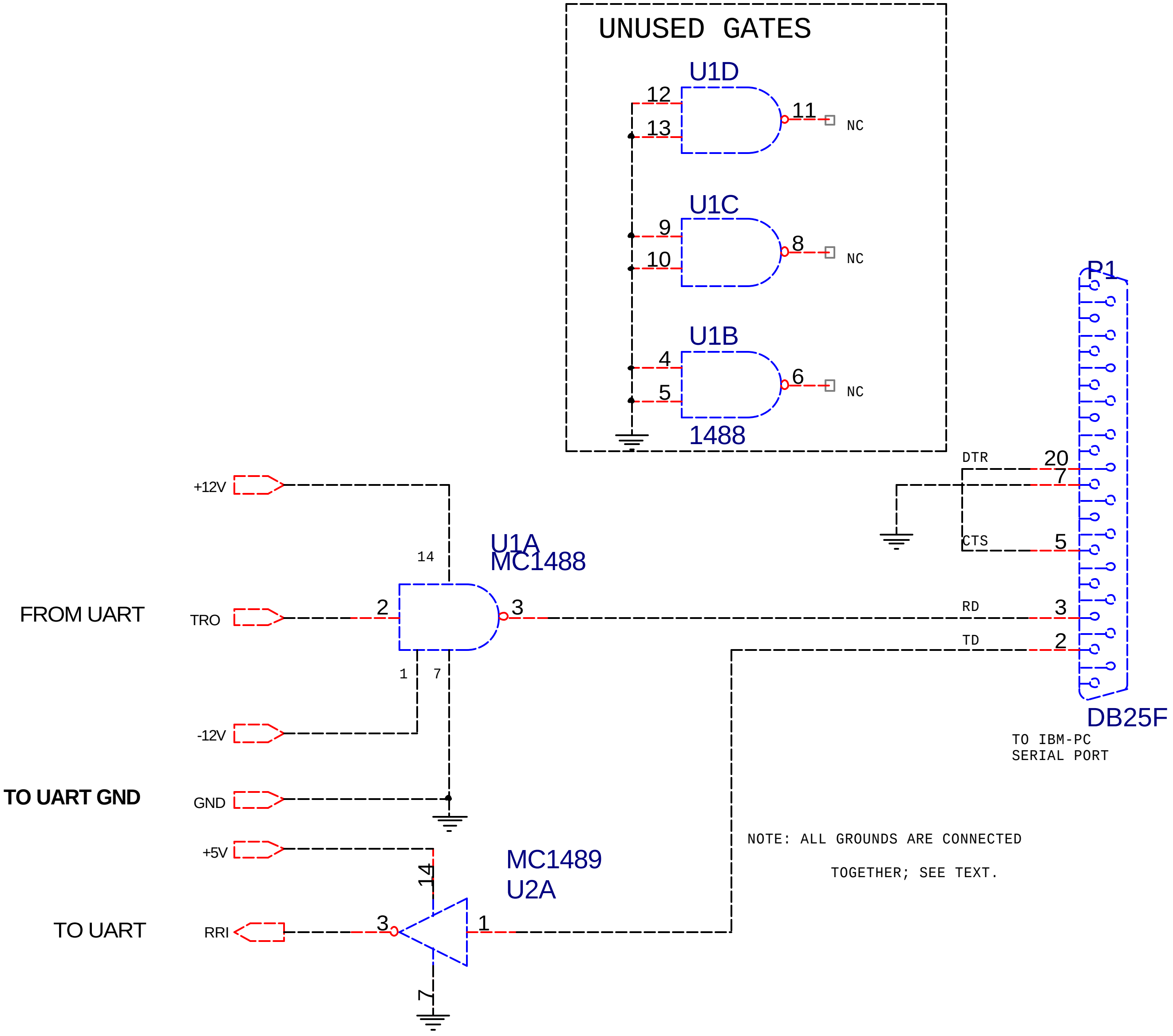

*Figure 1: RS232 Interface*

## *LABORATORY PROCEDURE:*

Name ________________ Sign-off _________

*PART I: CIRCUIT CHECKOUT PROCEDURE*

1. Build the circuit of Figure 1, but don't connect it to the UART or IBM-PC yet. Apply the three power supplies (+5, -12, and +12 Volts.)

**GROUNDING NOTE:**

Many new technicians have been known to scratch their heads when trying to figure out how to properly connect grounds in a circuit like this! The general rule for grounding is very simple: If the symbol for two grounds is the same, then they connect together; if it's not, they don't! Therefore, each of the three power supplies shares the ground buss in this circuit.

In some circuits, there will be two or more different ground potentials; these circuits will use different ground symbols for each type of ground in the system.

2. Test the transmit data circuit first. This is the circuit that will be converting UART data to RS232 levels on pin 3 of the interface. Here is a suggested method:

   a) Use a scope or DC voltmeter to monitor the Received Data pin on the RS232 connector (pin 3).

   b) Apply a TTL "1" to the "TRO" input. Measure the Received Data Voltage to make sure it is right.

   c) Do the same as step (b), but apply a TTL "0" instead.

3. Now test the receive data circuit. Remember, this circuit takes in RS232 levels and converts them to TTL. Devise your *own* procedure to test this part of the circuit. It will be similar but opposite as that from step 2. Explain in your laboratory report how your checkout procedure works.

4. If everything seems correct - congratulations! You're almost there. Now connect the UART "RRI" and "TRO" signals to the interface circuit. (And of course, power supply and ground.)

5. Make sure the UART parameters are 8,N,1 and 1200 BPS.

6. Have the instructor or an assistant check your circuit wiring before going on to the next step. An incorrect connection could damage the serial port of the laboratory computer!

*PART II: PC - UART INTERFACE*

7. Connect the DB25F to the rear of the IBM-PC. (The IBM-PC serial port has a male connector.)

8. Apply power to your circuit.

9. Boot the PC and start EZ-TERM. To start EZ-TERM, log onto Windows and double-click the EZ-TERM icon on the desktop.

**EZ-TERM COMMUNICATIONS SETTINGS**

EZ-TERM will remember the settings of the previous user, and these will appear in the title-bar of the EZ-TERM window. To alter the parameter settings, use the "Settings..." menu option (or the Port Settings button on the toolbar). Make sure the program is set to use the same communications port you are interfacing to!

10. The big moment is here. Try sending the letter "A" to the PC. (Now where did I leave that ASCII chart?!?) Load it carefully into the transmit data DIP switch and press DS (data strobe). Do you see a letter "A" on the screen (or a whole bunch of them, if your UART is sending continuous data?) If so, great! If not - don't panic. Go back and double-check your DIP switch settings and clock rate (it should be 16 X 1200 bps, right?)

11. Try pressing the letter "A" on the PC. (Upper case, please!) Did anything appear on the LEDs? Did the code look similar to the one you entered in step 9? If so, try some other codes. And don't forget to get your sign-off!

12. Can you tell what the difference between upper and lower case letters is in binary? (Compare 'a' *vs* 'A', 'b' *vs* 'B' to see if there is a pattern.)

*BUT IT DOESN'T WORK! (TROUBLESHOOTING)*

- *Nothing Works*...no lights, no nuttin'. Ahem..you guessed it: Power supply!

- *Works in one direction but not the other*: Probably means the clock and parameters are OK on the UART, but a data line (TRO or RRI) is not working. Use the 'scope to check it out. Follow the signal from the source to the destination to find out where it is lost.

- *Wrong data*: Yes, parameter settings. Check clock and parameter input switches.

*TIP: You can use the Hexadecimal display mode in EZ-TERM to see the binary equivalent of the characters you are transmitting to the PC. This way you can verify what codes the PC is receiving.*

## *Questions*

1. Explain step-by-step how data gets from the DIP switches on the UART circuit board to the display on the personal computer.

_______________________________________________________________________________

_______________________________________________________________________________

_______________________________________________________________________________

2. What is the purpose of the *terminal emulation* software used in this experiment?

__________________________________________________________________________________

__________________________________________________________________________________

__________________________________________________________________________________

3. What have you learned in this exp eriment?

__________________________________________________________________________________

__________________________________________________________________________________

__________________________________________________________________________________