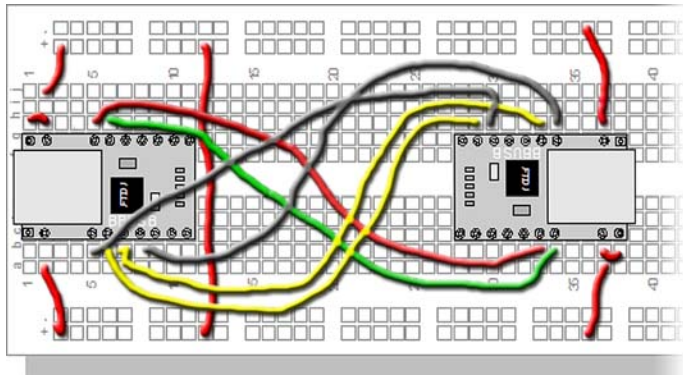
Chapter 11: Modem Line Hardware Experiments

Demo RTS/CTS and DTR/DSR lines

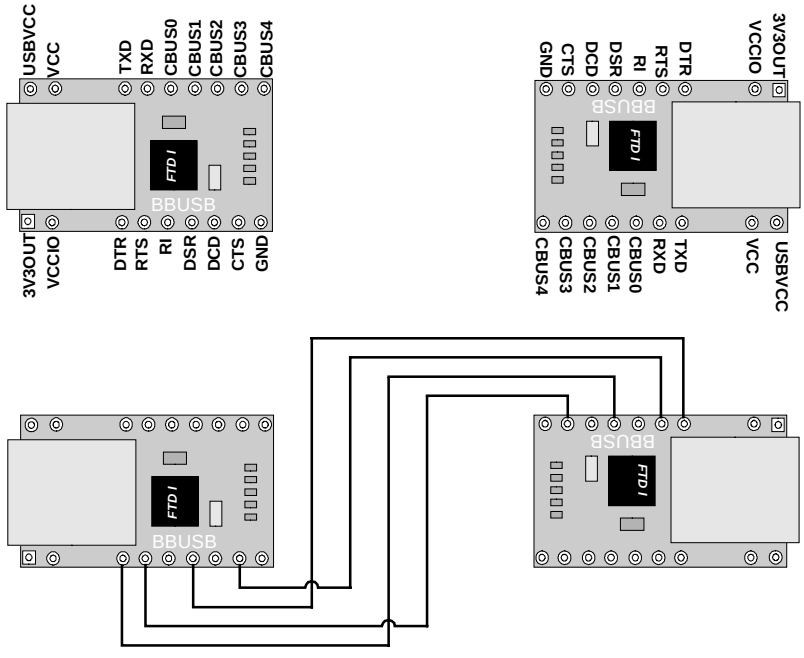
First lets use the modem line states sort of like they were intended. We will connect the RTS from one of our devices to the CTS of the other and visa versa. Also we will connect the DTR from one to the DSR. A real modem communication protocol would use these as discussed in the modem section to indicate readiness to send or receive serial data. We'll just push some virtual buttons and see the results on virtual LEDs. In the next section we'll use some real switches and LEDs.

We will simply observe that clicking the RTS button on one device causes the CTS LED to light up on the other and that clicking the DTR button on one device causes the DSR LED to light up on the other.

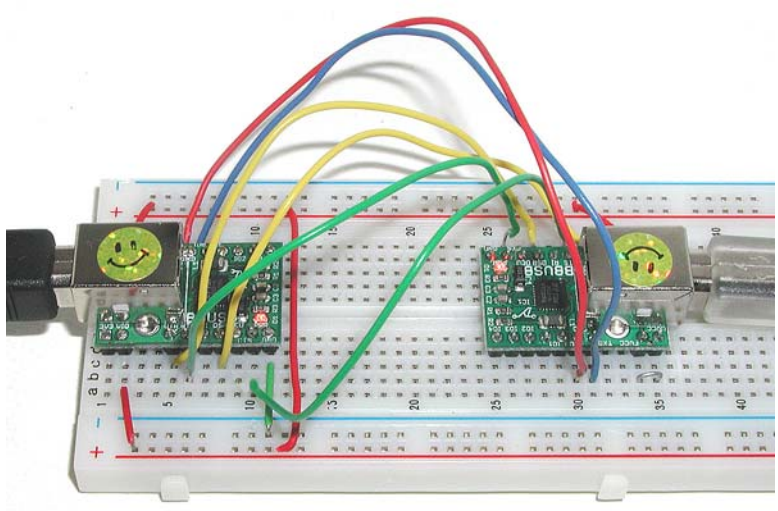
- Wire BBUSB 1 for +5V
  - Wire USBVCC to VCC
  - Wire VCC to breadboard +5V
  - Wire VIO to breadboard +5V
- Wire BBUSB 2 for +5V
  - Wire USBVCC to VCC
  - Wire VCC to breadboard +5V
  - Wire VIO to breadboard +5V
- Wire the RTS of each device to the CTS of the other.
- Wire the DTR of each device to the DSR of the other.
- The following illustrations show how to wire this experiment. Probably hard to follow in black and white:



- Possibly a more schematic approach will help?
- TODO SHOW THE POWER WIRING?



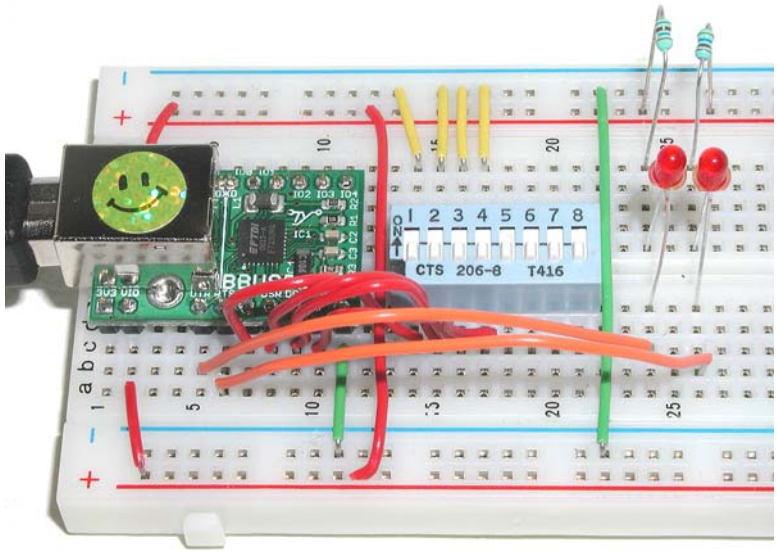
Okay, how about a photo to show the setup I used to do this test. I'm not a neat wirer, but the setup worked.



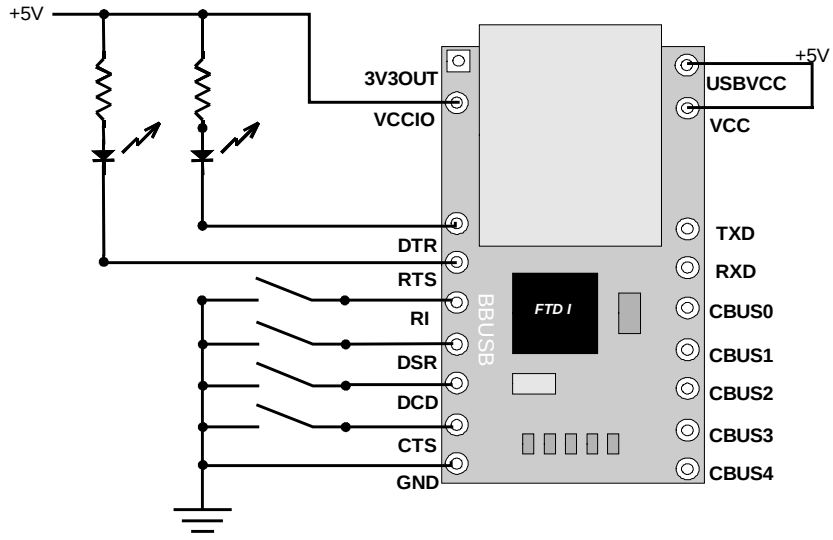
- Open two instances of the Developer Terminal.
- In one, open the left device port, in the other open the right device port.
- Click the RTS button in one and observe the CTS LED light up in the other.
- Click the DTR button in one and observe the DSR LED light up in the other.

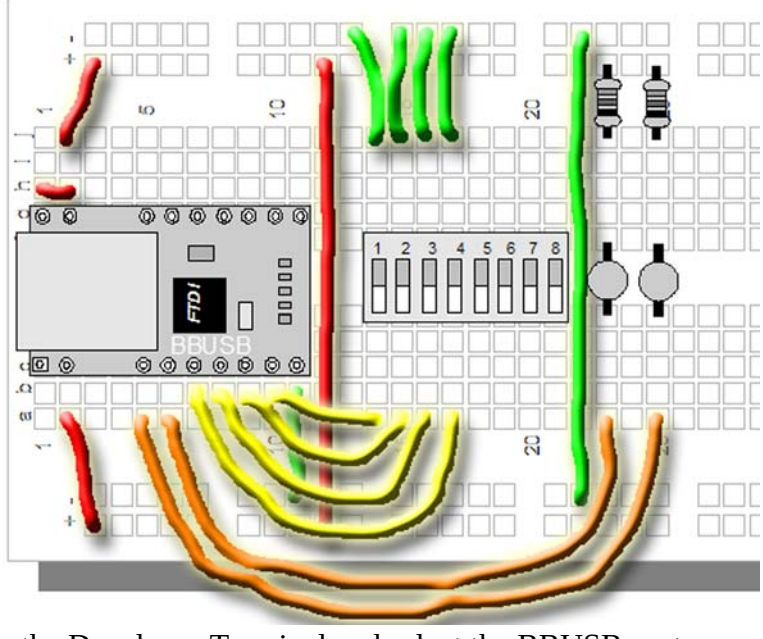
Off-label use – reading switches and lighting LEDs

Aw, to heck with all those modem lines, who uses modems anyway? But being able to control hardware by reading two on/off voltages or setting four on/off states, well now we are talking some real fun. For this experiment we will check switches for user input, and set LEDs for communicating with a user.



- Before wiring this up, unplug the USB cable. If you accidentally short power to ground your USB may just cut off with a message that it was drawing too much power, or your PC may crash. Guess how I know this.
- Wire the BBUSB for +5V
    - Wire USBVCC to VCC
    - Wire VCC to breadboard +5V
    - Wire VIO to breadboard +5V
    - Wire GND to breadboard GND
- Wire RI, DSR, DCE, and CTS to the off side of the first four switches on the 8 position DIP switch.
- Connect the on side of these switches to ground
- Connect DTR and RTS to the anodes of the LEDs.
- Connect the LED cathodes to +5V through the resistors.



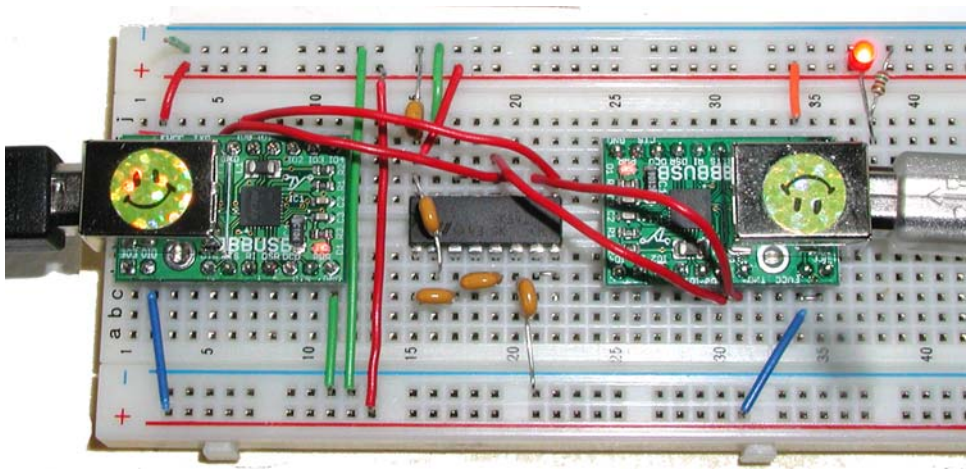


- Open the Developer Terminal and select the BBUSB port.
- Toggle the DTR and RTS buttons and note the LEDs turn on and off.

- Turn the four switches on and off and note the virtual LEDs on the Developer Terminal turn on and off.

RS232 Level Conversion

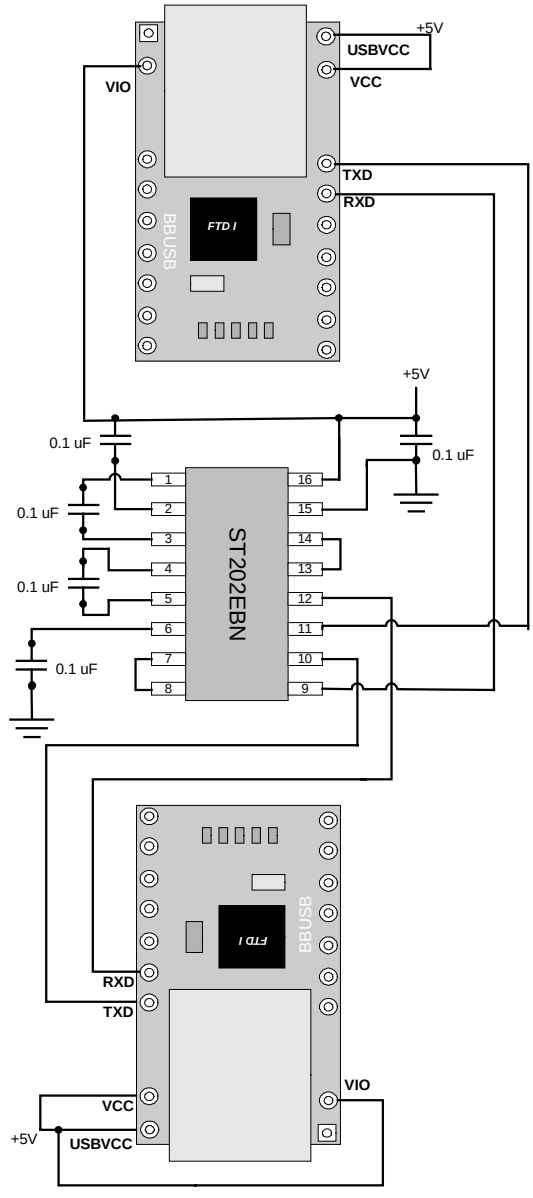
On first look this experiment might seem kind of stupid, and it certainly is a contrived way to demonstrate RS232 level conversions. Normally one would have two level converters one for each device on each end of a long cable, but here we are using a single converter and the devices are mere inches apart. But, this is a cheap way to demonstrate the principles and we do like cheap don't we?



And in addition to being stupid - it is hard. I wasn't paying adequate attention and had to use a scope to find a wiring error, so do be patient and follow the instructions (unlike me). So let's wire this sucker up.

- Place the two BBUSB and the ST202EBN on the breadboard more or less as shown.
- Look at Figure 2 of the ST202E/ST232E databook.
- Add the 5 0.1µF Caps as shown. Trim legs to get the caps near the board.
    - Cap between +5V and GND.
    - Cap between +5V and pin 2.

- Cap between pin 1 and 3.
- Cap between pin 4 and 5.
- Cap between pin 6 and GND.



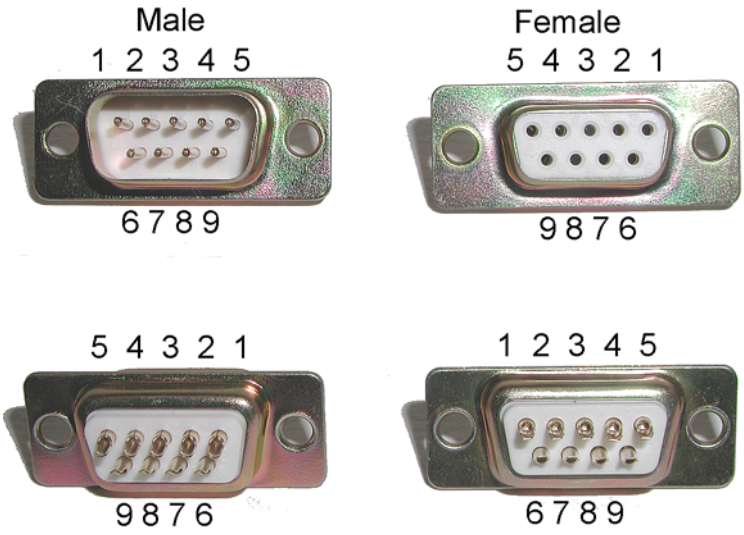
- Wire ST202EBN pin 7 to pin 8.
- Wire ST202EBN pin 13 to pin 14.
- Wire TxD of BBUSB 1 to pin 11 of theST202EBN.
- Wire RxD of BBUSB 1 to pin 9 of theST202EBN.
- Wire TxD of BBUSB 2 to pin 10 of theST202EBN.
- Wire RxD of BBUSB 2 to pin 12 of theST202EBN.
- Wire BBUSB 1 for +5V
    - Wire USBVCC to VCC
    - Wire VCC to breadboard +5V
    - Wire VIO to breadboard +5V
    - Wire GND to breadboard GND
- Wire BBUSB 2 for +5V
    - Wire USBVCC to VCC
    - Wire VCC to breadboard +5V
    - Wire VIO to breadboard +5V
    - Wire GND to breadboard GND
- Open two instances of the Developer Terminal
- Open BBUSB 1 in one instance and BBUSB 2 in the other instance.
- Okay, that WAS stupid, but you did learn all the principles didn't you?
- Did you also learn the wiring these things up correctly is harder than it appears?

.

USB via RS232 Cable – a Butterfly Experiment

You can go down to RadioShack and buy a USB to RS232 serial cable that has a USB connector on one end and a converter (often an FTDI device) with DB-9 connector on the other end. We are going to make something similar with the DB-9 part on the breadboard. We will use an AVR Butterfly which is **NOT** part of the projects kit. You can get one from www.smileymicros.com. You can, of course also use any RS232 serial device that only requires the Tx, Rx, and SG lines.

Signal	DB-9
DCD	1
Rx	2
Tx	3
DTR	4
SG	5
SR	6
RTS	7
CTS	8
RI	9

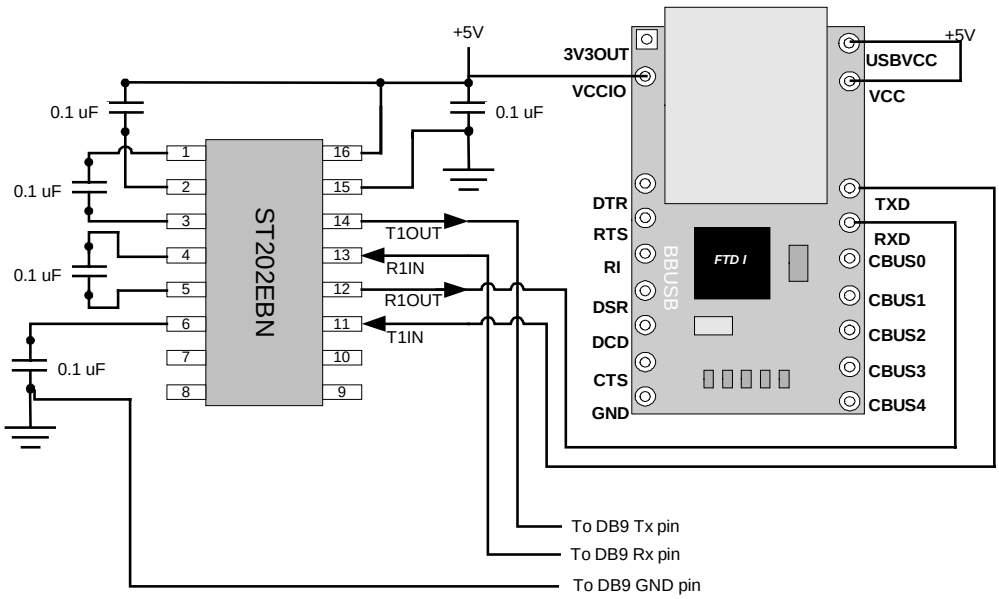


Solder Wires to back of male DB9 connectors

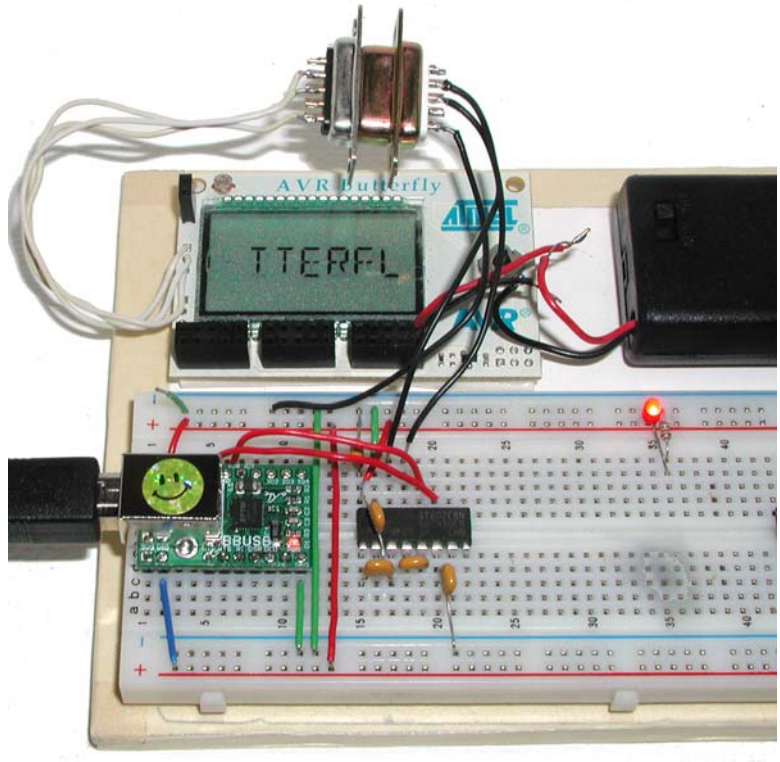
- This experiment is difficult to wire properly, I had to use a scope to find a stupid wiring error, so be careful and patient.
- For this experiment we will only use 3 of the 9 pins: Tx, Rx, and SG.
- We are using an ST202EBN which has two transmitters and two receivers. With this, we can do most common serial communications including having RTS/CTS handshaking. We cannot use the remaining modem lines

at the RS232 voltages levels since such chips would add a lot to the cost of the experimenter's kit and add nothing to the learning.

- Cut 3 3" pieces of 22 AWG wire.
- Strip about 1/8" insulation from one end and about 1/4" from the other.
- Solder the 1/8" end to the solder cups for Tx, Rx, and SG on the DB-9 connector.
- Remember that from the perspective of the PC and the external device the Tx and Rx are switched. This means that you connect the Tx line from the PC to the Rx line of the external device and visa versa.
- Wire up the ST202EBN level converter using the following illustrations. The ST202EBN has two TX and two Rx level converters and you can find the data sheet with the projects zip file. TODO PUT DATA SHEET



- Connect the DB9 male connector to the RS232 level converter as shown.
- Connect the DB9 male to the DB9 female of the Butterfly, which as said earlier is NOT part of the kit but cheap and available at – guess where?
- Notice that the power on the BBUSB side is separate from the Butterfly power, but that they have a common ground.


AVR Butterfly
TTERFL