HACKING *the* CABLE MODEM

WHAT CABLE COMPANIES DON'T WANT YOU TO KNOW

DERENGEL

NO STARCH PRESS

17

BUILDING A CONSOLE CABLE

The device shown in Figure 17-1 is an RS-232–to–TTL converter board, designed to allow a PC with a serial (RS-232) port to communicate with a device that has a console (TTL) port. External converters such as this are common, and you can purchase one from many online electronics stores. Or, with the right parts, you can build your own inexpensive RS-232–to–TTL converter, known as a *console cable*.

The Console Port

Many embedded devices (such as switches, routers, cable modems, and so on) have an internal communication port known as a *console port*. This type of port is typically used for configuring the device and issuing commands with root-level access. If the device is offline, this port can also be used to reconfigure the device locally. However, if it is online, other administration protocols can also be used, such as telnet or rlogin.

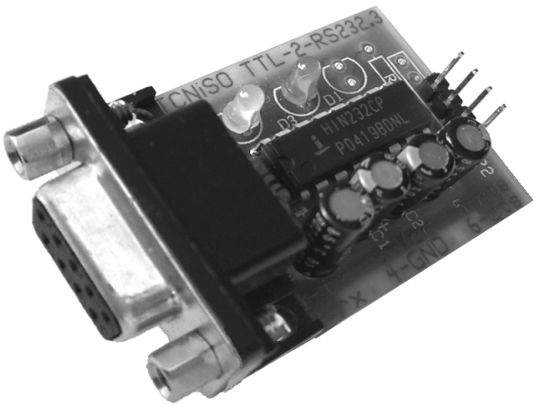

Figure 17-1: A professionally developed RS-232 console port

Many cable modems have a clandestine console port left over from debugging during the manufacturing process. This port can sometimes be utilized to access the device's bootloader program or operating system, allowing the user to change many of its internal settings (MAC address, serial number, and so on) or its firmware, and/or execute system commands. Because having the ability to communicate using this port may by itself be enough to hack a cable modem, it is important to know how to communicate using this type of port.

What Is TTL?

Transistor-Transistor Logic (TTL) is an interface often used to communicate between integrated circuits. If a cable modem has an unused console port, that port will most likely be accessible using a TTL-compatible interface. While your computer probably does not have ports that support TTL signals, you can build a port converter from scratch or purchase one from many electronics stores.

The easiest way to connect your computer to a TTL console port is with a serial (RS-232 or DB9) port on your computer. If your computer does not have such a serial port, you can purchase a USB-to-serial adapter for around $20.

The cable modem's TTL port will not usually have a connector, so you will most likely have to build one and solder it in. Then, once your computer's serial port is connected to the modem through the RS-232 converter, you can communicate with it through the port using any terminal emulation software, such as HyperTerminal or EtherBoot.

Examining the Schematic

Figure 17-2 shows you how to properly convert an RS-232 signal to TTL levels.

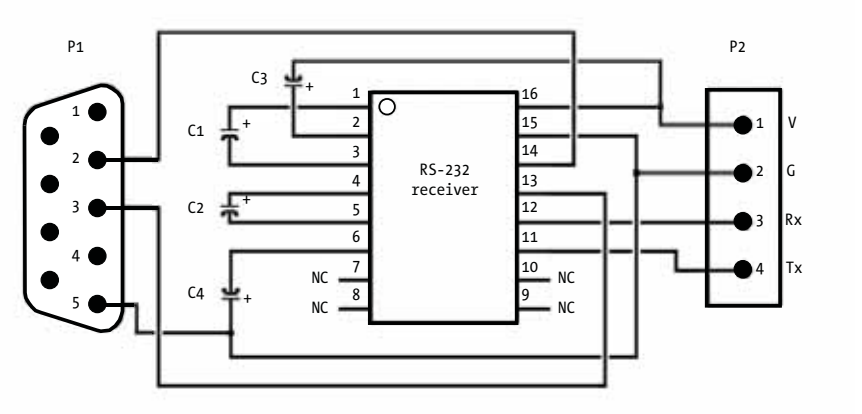

Figure 17-2: Schematic of circuit to convert RS-232 to TTL

Components P1 and P2 are the input/output connectors. P1 represents the end of a serial port or serial cable; the numbers inside it correspond to specific pins of this port. Often, if you observe the end of a serial cable, you will see an indentation or marking that signifies the first pin.

P2 represents the four-pin TTL console port. Unlike the serial port, its pins may be in no specific order. Instead, its pins are labeled by type: *V* represents *voltage* (usually 3.3 or 5V); *G* represents *ground*, *Rx* represents *receive*, and *Tx* represents *transmit*.

Components C1 through C4 are capacitors, rated from 0.1 to 10µF at 50V. The capacitors should be facing in the direction shown in the schematic, in which a small plus sign (+) indicates the way that the positive side of the capacitor should face. However, not all capacitors are labeled the same way, so you should always check the datasheet of the capacitor from the manufacturer. If a capacitor is placed incorrectly, the entire circuit may not work properly.

The integrated circuit, shown in the middle of Figure 17-2, must be a compatible 16-pin DIP RS-232 driver/receiver chip. The NC label means *no connection* and tells us that certain pins should not be connected to anything.

NOTE *Many semiconductor companies, such as MAXIM and Intersil, produce chips that are compatible with this design. However, if you use another package type or manufacturer, read the device's datasheet and compare its input/out pins to this schematic.*

How to Build a Console Port

The following instructions describe how to build your own console port from scratch. If you are a computer junkie like me, you may already have all the parts needed. For example, the most important part you need is a RS-232–to–TTL integrated circuit chip, which you might find in an old serial mouse or smartcard programmer. I suggest you go through your old computer junk and look for devices that use a serial port, and then open them to see if they have such a chip inside.

Step 1: Gather the Parts

The first obstacle you need to overcome is the distance between your computer's RS-232 port and your cable modem. If you're on a budget, you could use a female-to-male DB9 serial cable (three to six feet long) and simply cut off the male end, exposing the nine individual wires. These cables are very common.

A better (and more expensive) method is to use a special one-sided DB9 serial cable (shown in Figure 17-3) that is designed for electronic projects. This type of cable has pins that are color-coded to indicate the pin numbers. (In contrast, a generic serial cable may not have color-coded pins, or the colors may be inconsistent.) If you do not know the pin numbering on your cable, use a standard voltage meter to find them.

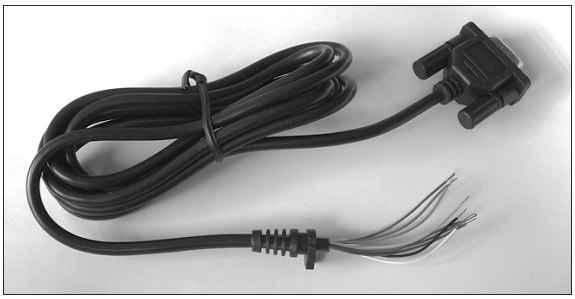

Figure 17-3: Serial DB9 "project" cable

In order to build your converter circuit, you will need something strong to hold your device together and allow you to easily solder joints. For this purpose, I recommend either a general-purpose IC PCB or a prefabricated punch board, both of which can be purchased at Radio Shack for under $5. The general-purpose IC PCB has predrilled holes and metal contacts which are easy to solder onto, though I recommend the prefabricated punch board shown in Figure 17-4, which you can easily cut into any shape you want.

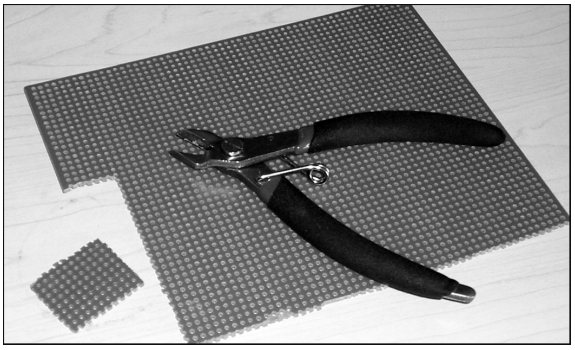

Figure 17-4: Prefabricated punch board

The most important part is an RS-232 driver/receiver interface circuit that outputs to TTL levels. I recommend either a MAX232CPE from www.maxim-ic.com or an HIN232CP from www.intersil.com.

You will also need four 1µF capacitors. I recommend purchasing several 50V 1µF radial electrolytic capacitors like the ones shown in Figure 17-5.

Finally, you will need some insulated wire for connecting your converter to the modem. I recommend wrap wire from Radio Shack.

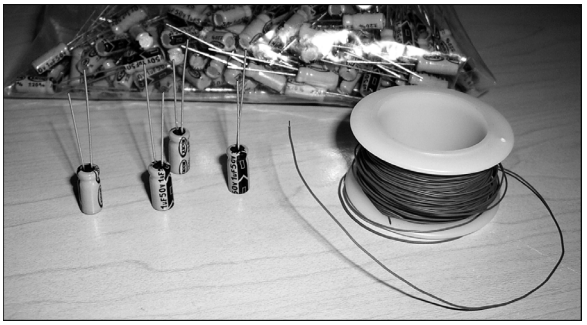

Figure 17-5: 50V 1µF capacitors and wrap wire

Step 2: Gather the Tools

The most important tool you will need in order to actually construct the converter is a low-temperature soldering iron, rated 30 to 40W. You will also need two or more ounces of rosin core solder and a pair of small wire clippers. Figure 17-6 shows all the tools you will need.

Figure 17-6: Tools you need to build a console cable

Step 3: Put the Pieces Together

Once you have acquired all the necessary parts and tools, you can begin to assemble your own console cable.

1. Use your clippers to cut a piece out of the prefabricated punch board that is 8 holes wide and around 14 holes long. This smaller board will be the basis for your converter circuit. Insert the pins of the RS-232 driver/receiver interface chip into the middle of this board, making sure to leave a gap of least two holes on every side. (You will sometimes need to squeeze and straighten the pins with your fingers in order to get them to fit in the holes properly.)

2. Insert one of the capacitors in the holes next to pins 1 and 3 of the interface chip, making sure that the positive end of the capacitor is in the hole adjacent to pin 1 of the chip. (If you do not know which pin represents number 1, look for the pin next to the circular indentation on the chip; however, this may not be the case with all chips, which is why it is always important to check the manufacturer's datasheet.)
3. After you place the two leads of the capacitor through the holes, bend them so that they lay flat next to the pins from the chip, and then apply solder to connect the lead of the capacitor to the pin of the chip. (You may want to use your clippers to cut off the part of the capacitor lead extending past the solder point.)
4. Repeat steps 2 and 3 with the capacitor for pins 4 and 5 of the circuit chip. Again, the positive end of the capacitor should be adjacent to pin 4.
5. Place the negative end of the third capacitor next to pin 6 of the chip and the other end at a hole that is past pin 8. We will use this hole as a common ground in our circuit.
6. The last capacitor needs to be connected to pin 2 (the positive side) and the shared voltage line of your circuit. I recommend placing the capacitor's leads through two holes just above the top of the chip and then bending the positive lead to connect pin 2 and the negative lead to connect pin 16 (the input voltage of the chip).

Once you have finished putting these pieces together, your device should look similar to the one shown in Figure 17-7.

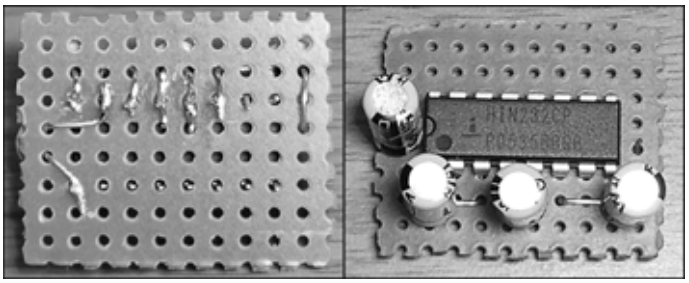

Figure 17-7: Building the circuit

Step 4: Connect the RS-232 Cable

The next step is to take the end of a DB9 serial cable (also known as an RS-232 cable) and connect it to your RS-232–to–TTL device.

1. If you have a regular RS-232 serial cable, cut off one end and expose the leads of the individual wires inside the cable.
2. Using an electronic multimeter, find and mark the wires that correspond to pins 2, 3, and 5 at the female end of the DB9 connector. Pin 2 is used to receive data to your PC, pin 3 is used to transmit data from your PC, and pin 5 is used as ground.

3. With your serial cable ready, solder pin 2 from the serial cable to pin 14 of the chip. I often find it helpful to thread the thin wire through a couple of the spare holes, so that tension in the cable will not accidentally break off the soldered connection.
4. Repeat this step with pin 3 from the serial cable, and solder it to pin 13 of the chip.
5. Pin 5 from the serial cable is the shared ground; solder this to the solitary capacitor lead (see "Step 3: Put the Pieces Together" on page 163), but leave enough room to solder more connections here later.

Step 5: Connect the TTL Lines

The next step is to connect four pieces of wire to the integrated circuit, as shown in Figure 17-8. These four wires will be used to connect your cable to the console port inside the modem.

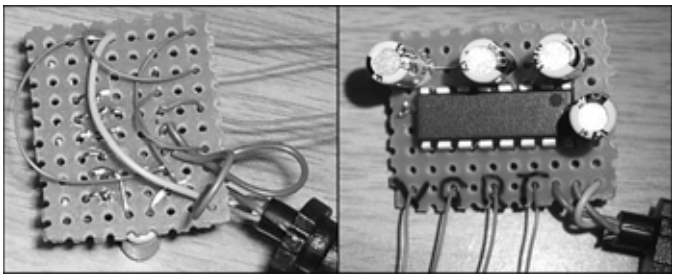

Figure 17-8: Finishing the serial cable

1. Using your wrap wire, cut four pieces (six to eight inches each) and one smaller piece (two to three inches) and strip off the ends, exposing the metal inside.
2. Solder a long piece of wire to pin 16 of the chip (this is the voltage pin of the chip).
3. Solder the small piece of wire from pin 15 to the shared ground connection (see "Step 4: Connect the RS-232 Cable" on page 164).
4. Solder another long piece of wire to your shared ground connection.
5. Solder your last two long pieces of wire to pins 12 and 11 of the chip.
6. Using a marker pen (like a Sharpie), mark the top of your board with the symbols *V* (voltage), *G* (ground), *R* (receive), and *T* (transmit) to help you remember and recognize the functions of each long piece of wire.
7. Take the wire that you soldered to pin 16 on the chip and put it through a hole close to the *V*.
8. Put the wire that is connected to your mutual ground through the hole marked with a *G*.
9. Put the wire connected to pin 12 through the hole marked with an *R*.
10. Put the wire connected to pin 11 through the hole marked with a *T*.

Your finished cable should now look like the one shown in Figure 17-9.

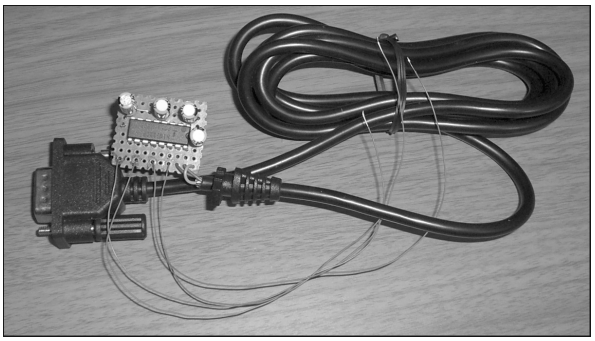

Figure 17-9: The finished RS-232 console cable

Your finished RS-232–to–TTL console cable should now be ready for use. If you wish to strengthen the cable so that it may last longer, use a lot of hot glue to make a strong protective layer around your board, the wires, and the places where you soldered.

To use your new console cable, connect the female end of the DB9 connector to the COM1 serial port on the back of your computer, and connect the four loose wires to the console port of your target device (in this case, your cable modem).

Step 6: Connect the Cable

It can often be very difficult to connect a console cable to your cable modem because it can be so hard to find the port to which you need to solder your four wires. The four wires from your console cable should be connected to the console port as follows. The wire from your converter board marked with a *V* needs to be connected to a 3.3V or 5V positive power source. The wire marked with a *G* needs to be connected to any grounded connection on the target board. The wire marked with an *R* needs to be connected only to the data-in pin of the console port. And finally, the wire marked with a *T* needs to be connected only to the data-out pin of the console port.

For further help on connecting your console cable to your modem, download TCNISO Video #1 from www.tcniso.net/Nav/Video. This video shows you how to open your modem, solder the cable to the PCB, use the EtherBoot software to communicate with your modem, and then change the firmware.

NOTE *Chapter 18 contains pictures and diagrams of the locations of the console port in many popular cable modems, such as the SB4*xxx *series.*

Search for the Console Port

When you open your modem to search for a console port, look for an array of four metal pins sticking up from the board or for four solder pads with nothing connected to them. Unfortunately, the pins on a console port can be arranged in any order, so you may need to use a multimeter and some trial and error to find the correct mapping or identity of the pins.

If you find what appears to be a console port, use your multimeter to test the pins. The ground pin should have perfect continuity to the metal plate on the back of the modem or to the metal of the tuner. With the device plugged in, use your meter to find the voltage pin, which must maintain a steady 3.3 or 5V. The Tx pin of a console port should be at about ±3V, while the Rx pin should remain at 0V.

A console port might be made up of just the receive (Rx) and transmit (Tx) pins, as is the case with the SB3100 and SB4*xxx* series cable modems. If this is the case, you will need to connect the ground and voltage of your console cable to the modem and then find the Rx and Tx connections by trial and error.

Some time ago, I had an SB3100 SURFboard cable modem whose console port did not function correctly. The port would transmit data to my computer, but I was unable to send data back to the Rx port. I believed that the physical port itself was damaged or defective. After referencing the datasheet for the chipset, I decided to manually solder the Rx wire of my console cable directly to the chipset. This worked, and Figure 17-10 is a picture taken shortly after this was done. I used hot glue to keep the wire from breaking off. This is a good example of how to manually find the console port.

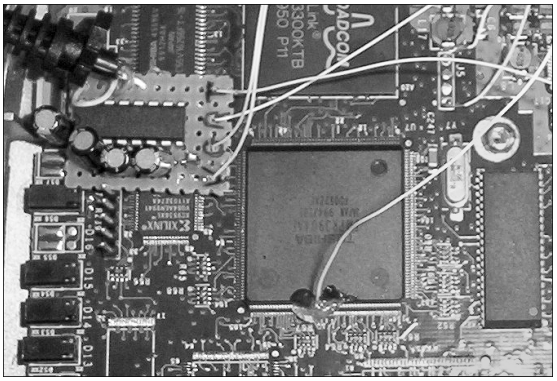

Figure 17-10: An SB3100 modem chipset with the Rx pin connection

Step 7: Test Your Console Cable

With your new console cable connected properly from your PC to your cable modem, you next need to set up and run terminal emulation software. You can use HyperTerminal (which comes standard on most Windows PCs) or EtherBoot (Figure 17-11). Once your software is running, it is usually necessary to reboot the modem, which will cause startup data to be displayed in your terminal software's console window.

When using HyperTerminal, you can create a new connection using the COM1 port and then configure the properties for this connection according to your device. Settings such as the bits per second (baud rate) are very important because an incorrect value can result in garbage data being seen in the console window. You will almost always need to set the flow control to *None*. (If you don't know your device's proper settings, you will have to use trial and error to find them.)

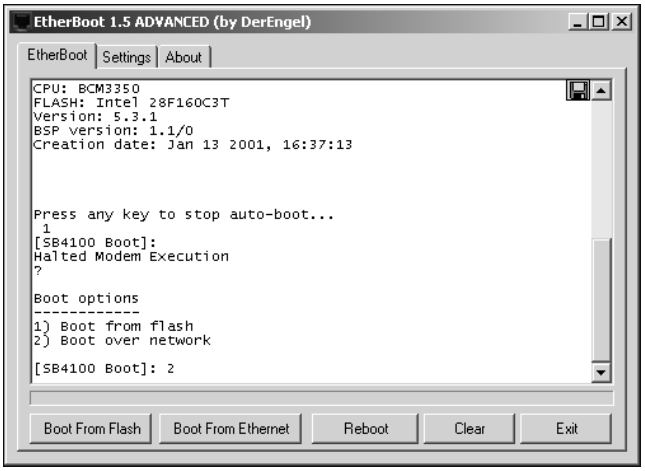

Figure 17-11: EtherBoot successfully connected to the console port

EtherBoot is a terminal emulation program that is customized for cable modems; for information about where to download this program, please see Chapter 13. You simply select your modem's model name in the Settings menu to quickly configure the software. This software also includes many additional features, such as the ability to boot firmware on the fly. (See Chapter 13 for more on EtherBoot.)

When you plug in your cable modem with your terminal software running, output such as that shown in Figure 17-11 may be displayed in your software's console window. Output like this tells you that the Tx connection of your console cable is working correctly. If you can type characters into your console window and read them, then the Rx connection is also working correctly. If, however, random ASCII garbage is displayed, your baud rate may be set incorrectly, or your console cable may not be properly grounded.

Limitations of a Console Port

Many cable modems have console ports that allow you to do low-level operations, like booting firmware or changing the MAC address. Some, however, have the entire console port disabled or have the Rx line disabled (which prevents a user from sending data). These restrictions are usually set via the embedded firmware.

A good example of this limitation is implemented in the SB5100 SURFboard modem. Normally, when a user tries to communicate with the SB5100 using a console cable, data will be displayed to the console window; however, the user cannot send data back to the modem. The good news is that there is a hack available to permanently enable the console port on this modem. You can use the Blackcat firmware modification tool (see Chapter 15) to program a new bootloader into the modem (at the beginning portion of the firmware), which will then allow you to use a console cable to communicate with the SB5100.