computers are bad

You are receiving this facsimile because you signed up for fax delivery of this newsletter. To stop delivery, contact Computers Are Bad by email or fax.

https://computer.rip - me@computer.rip - fax: +1 (505) 926-5492

2022-01-24 the smart modem

I think I've mentioned occasionally that various devices, mostly cellular modems, just use the Hayes or AT command set. Recently I obtained a GPS tracking device (made by Queclink) that is, interestingly, fully configured via the Hayes command set. It's an example of a somewhat newer trend of converging the functionality of IoT devices into the modem baseband. But what is this Hayes command set anyway?

Some of you are no doubt familiar with the "acoustic coupler," a device that has two rubber cups intended to neatly mate with the speaker and microphone of a telephone handset. The acoustic coupler allowed a computer modem to be connected to the telephone system via audio instead of electrically, which was particularly important because, pre-Carterfone, nothing could be connected to the telephone system that was not leased from the telco. Acoustic couplers were also just convenient, as back in the days when all equipment was leased from the telco phones were fairly expensive, most houses did not yet have a panoply of telephone jacks, and so it was just generally handy to be able to easily use a normal phone with computer modem without having to swap around cabling.

Unfortunately, this scheme had a major limitation: the computer interacted with the telephone just like you would, via audio. The computer had no way, though, of taking the phone on or off hook or dialing. That was all up to the user. So, you'd pick up the phone, dial a number, and then set the phone down on the acoustic coupler. When you were done, you would take the phone off of the coupler and hang it back up. Besides being a bit of a hassle and sometimes prone to mistakes, this effectively ruled out any kind of automatic or scheduled modem usage.

Through the '70s, modems capable of automatic dialing and on/off hook were available but were expensive, large machines intended for commercial-scale use. For example, they were somewhat widely used by retail point of sale systems of the era to send regular reports back to corporate headquarters for accounting. For the home computer enthusiast, there were essentially no options, and among other implications this ruled out the BBS ecosystem that would emerge later since there was no way for a computer to automatically pick up the line.

Everything changed in 1981. Actually, the first fully computer-controlled modem came somewhat earlier, but because it was designed specifically for S-100 computers (like the Altair) and later Apple II, its popularity was limited to those platforms. Hayes, the same company that developed this early internal modem, released the Hayes Smartmodem in '81--which truly started the PC modem revolution. The basic change from their earlier internal modems was that the Smartmodem interfaced with the host computer via serial. RS-232-esque-ish serial ports were by this time ubiquitous on microcomputers, so the Smartmodem could be used with a huge variety of hardware.

It might be surprising that a modem that allowed programmatic control of the hook and dialing took so long to come around. It might be more obvious why if we think about the details of the modem interface to the host PC. The task of a modem is, of course, to send and receive data. In order to do so, modems have traditionally acted like transparent serial channels. In other words, modems have behaved as if they were simply very long serial cables between two computers. Whatever data was sent to the modem it transmitted, and whatever data it received it returned on the serial interface.

We could thus refer to the serial connection to the modem as being the data plane. How is the modem commanded, then? Well, originally, it wasn't... the user had to handle all aspects of call control manually. To bring about automatic call control, Hayes had to come up with a command set for the modem and a way to send those commands. Hayes solution is one that vi users will appreciate: they created two modes. A Hayes Smartmodem, in data mode, acted like a normal modem by simply sending and receiving data. A special escape sequence, though, which defaulted to "+++", caused the modem to change to command mode. Once in command mode, the computer could send various commands to the modem and the modem could reply with status information. The modem would switch back to data mode either after an explicit mode switch command or implicitly after certain connection setup commands.

All commands to a Hayes modem began with the letters "AT". There are a few reasons for this. Perhaps most obviously (certainly to any vim users), the use of two distinct modes creates a huge opportunity for "mode errors" in which the modem is somehow not in the mode that the software controlling it thinks it is. Prefixing all command strings with "AT" serves as an additional check that a line of text is intended to be a command is not actually data errantly sent during command mode, which might cause the modem to take all kinds of strange actions. Second, AT was used for automatic baud detection and clock recovery in the modem, since it was a known bit sequence that would be sent to the modem after the modem first powered on and before it was used to make a call.

It's because of this "AT" prefix, which in principle stands for "attention," that the Hayes command set is commonly referred to as the AT commands. If either Hayes or AT rings a bell, it will be because the influence of the Hayes Smartmodem on the computer industry has been incredibly long lasting: essentially all telephone network modems, whether landline or cellular, continue to use the exact same Hayes interface. In most cases, the operating system on your smartphone is, as we speak, using the Hayes command set to interact with the cellular baseband. If you buy an LTE module for something like an IoT application, you will need to send it Hayes commands for setup (under Linux the ModemManager daemon is responsible for this background work). If you use a USRobotics parallel telephone modem, well, you will once again be using the Hayes command set, but then that's less surprising.

Let's take a quick look at the Hayes commands. The format of them is somewhat unconventional and painful by modern standards, but keep in mind that it was intended for easy implementation in '80s hardware, serial interfaces were slow back then, and in general it is designed more for economy than user-friendliness. On top of that, it's been hugely extended over the years since, meaning that the awkward "extension" commands are now very common.

A Hayes command string starts with "AT" and ends with a carriage return (line feed may or may not be used depending on configuration). In between, a command string can contain an arbitrary number of commands concatenated together--there is no whitespace or other separation, rather the command format ensures that it is possible to determine where a command ends. You can imagine this has become a bit awkward with extension commands and so, in practice, it’s common to only put one command per line except for rather routing multi-step actions.

The basic commands consist of a single capital letter which is optionally followed by a single digit. Most of the time, the letter indicates an action while the digit indicates some kind of parameter, but there are exceptions. Some commands take an arbitrary-length parameter following the command, and some commands accept a letter instead of the one trailing digit. Actually even the original Hayes command set is so inconsistent that it’s hard to succinctly describe the actual syntax, and now it’s been added on to so many times that exceptions outweigh the rules. It might be easier to just look at a few examples.

To do perhaps the most obvious thing, instruct the modem to go off hook and dial a telephone number, you send “ATDT” (D=Dial, T=Touch-Tone) followed by a string which specifies the phone number... and can also contain dialing instructions such as pausing and waiting for ringback. For example when dialing into a PABX that uses extensions, you might use “ATDT5055551234@206;”. This tells the modem to dial (D), touch-tone (T), 505-555-1234, wait until ringback (@), dial 206, then stay in command mode (;). Without the semicolon, the D command usually causes an implicit switch to data mode.

Answering a call is simpler, since there are fewer parameters. The “A” command is answer. The command string would sort of technically be “ATA0” since A ostensibly conforms to the “one letter one digit” convention, but when the digit is 0 it can be omitted.

But wait... how would the computer know that the modem is “ringing” in order to answer? Well, for that you’ll have to jump back to the post on RS-232, and study up on why the Hayes Smartmodem used a 25-pin connector. There’s just a dedicated wire to indicate ringing, as well as a dedicated wire to indicate when the modem is ready to move data (i.e. when a data carrier is present). The serial interface in the computer was expected to expose the state of these pins to software as needed.

Some of you may remember that, in the days of dial-up, it was common to hear the modem dial and negotiate the data connection aloud. This too dates back to the Hayes Smartmodem, and it’s somewhat related to the reason that fax machines usually provide a handset. If you misdial or there is a problem with the destination phone number or one of a number of other things, you may get an intercept message or someone answering or some other non-modem audio upon the call connecting. The Smartmodem featured a speaker to allow the user to hear any such problems, but of course few users wanted to listen to the whole data session. The Hayes “M” command allowed the host computer to set the behavior of the speaker, and “ATM1” was commonly sent which caused the modem to enable the built-in speaker until a data carrier was established, at which point it was muted.

The Hayes Smartmodem also included a number of registers in which configuration could be stored in order to affect the behavior of later commands. For example, the duration of a standard dialing pause could be adjusted by changing the value in the register. The “S” command allowed for selecting a register (e.g. ATS8 to select register 4), and the “?” and “=” commands could be used to query and set the value of a register. “=” of course took an argument, and so “ATS8=8” could be used to set the pause duration to 8 seconds. This might look like one long command but it’s not, we could just as well send “ATS8” followed by “AT=8”. The = is a command, not an operator.

As modems became faster and more capable and gained features, the Hayes command set gained many additions and variants. While the core commands remain very consistently supported, the prefixes “&”, “%”, “", and”+" are all used to indicate various extended commands. Some of these are defined by open standards, while others will be proprietary for the modem manufacturer. For example, the GSM standard specifies extended Hayes commands useful for interacting with cellular modems. For example, “AT+CSQ” can be used to ask a cellular modem for the current signal strength (RSSI). The “+” prefix is, in general, used for ITU-standardized additional commands, and “+C” used for commands related to cellular modems. You’ll see these prefixes very frequently today, as the Hayes command set is more and more seen in the context of cellular modems rather than telephone modems.

Of course, “+CSQ” being a command seems to violate the syntax I explained earlier for Hayes commands, and vendor proprietary commands frequently take this much further by introducing multi-parameter commands with parameter separators and all types of lengthier command names. For example, for a personal project I wrote software around a Telit LTE module that made use of the command string “AT#CSURV” (note non-standard prefix “#”). This command causes the modem to search for nearby cells and return a listing of cells with various parameters, which is useful for performing site surveys for cellular network reliability.

Many modern cellular modems have GPS receivers built-in, and it’s possible to use the GPS receiver via Hayes commands. On the Telit module, a command string of “AT$GPSACP"causesthemodemtoreturnthecurrentposition,whilethecommandstring"AT$HTTPGETSTSEED=1,2199” (note two parameters) can be used to command the embedded GNSS module to load AGPS data from an HTTP source (the details of AGPS will perhaps be a future topic on this blog).

Brief tangent: some of you may be aware (perhaps I have mentioned it before?) that dialing emergency calls on GSM and LTE cellphones is, well, a little weird. Much of that is because the GSM specifications have built-in support for emergency calling, independent of phone numbers, that is intended to allow cellular phones to present a consistent emergency calling method regardless of the dialing conventions of a country/area the user might be roaming in. The exact commands are unstandardized, but on the Telit module “AT#EMRGD” initiates an emergency call (note that no phone number is specified) while “AT#EMRGD?” (it is a common convention in extended AT commands for a trailing “?” to change the command to a status check) causes the modem to report which phone numbers the GSM network has indicated should be used for different types of emergency calls--chiefly for display to the user. This is why dialing common international emergency numbers like 999 and 110 on a US cellular phone still results in a connection to 911--in actuality no dialing happens at all, when the dialer app determines that the number entered appears to be an emergency call it instead issues an AT command with no phone number at all. Part of the reason for this is due to enhanced GSM features for position reporting, which relate to what is called “e911” in the US and provide essentially a basic, slow data channel between a cellular phone and a PSAP that can be used by the phone operating system to report a GPS position to the PSAP. There are, of course, a half dozen AT commands on most cellular modems to facilitate this [1].

Now keep in mind that all of these commands happen over a channel that is also intended to send data. So, after dialing a call or by issuing the command string “ATO” (go online) further data sent over the serial connection will instead go “through” the modem to the other end. In practice, though, mode switching introduces a set of practical problems (not least of which is having to make sure the escape sequence “+++” does not appear in data) and so most modern modems actually don’t do it any more. Instead, the Hayes protocol serial connection is usually used purely for modem commanding and a separate data channel is used for payload.

This is clearest if we look at the most common modern incantation of Hayes commands, a cellular modem connected to a host running Linux. Traditionally, ModemManager would issue a set of commands to the modem to set up the connection after which it would place the modem into data mode and then trigger pppd to establish a ppp connection with the modem serial device. In practice, most cellular modems today are “composite devices” in some sense (i.e. present multiple independent data channels, whether physically or as a virtual product of their driver) and appear as both a serial device and a network interface. The serial device is for Hayes commands, the network interface is, well, a plain old network interface, which makes network setup rather easier than having to use PPP. There are various ways that this happens mechanically; in the case of USB modems it is usually by presenting a composite USB device that includes some type of network interface profile like CDC Ethernet [2].

In fact, a lot of modems don’t just present a serial interface and a network interface... it’s not unusual for modems to present several. One will be for Hayes commands, but there’s often a second to be used as a dedicated channel for PPP over serial (in case a different method of network connection isn’t used) and then a third dedicated to GPS use. Since applications often want regular (unsolicited) updates from the GPS module and it’s a bit silly to have to constantly poll via Hayes command or switch modes around, it’s common for LTE modems to allow the host to issue a Hayes command that enables unsolicited GPS updates, after which they continuously generate GPS fix messages on a dedicated channel. These are usually in NMEA format, a widespread standard for GNSS information over simple serial channels that was originally developed to allow a single GNSS receiver on a boat to disseminate position information to multiple navigation devices. Yes, specifically a boat--NMEA is the National Marine Electronics Association, but they came up with a solid standard first and everyone else has copied it.

Despite the partial shift away, Hayes commands have a lot of staying power due to their simplicity. Some devices are going to the direction of using *more* Hayes commands, since it can actually eliminate the need for any “data channel proper” in some cases. Many LTE modems oriented towards IoT or industrial use provide extension Hayes commands that perform high level actions like “make a POST request”. The modem implements HTTP internally so that developers of embedded devices don’t have to. The Telit modules even support HTTPS, although setting up the TLS trust store is a bit of a pain.

The latest hotness in cellular modules is the ability to load new functionality at runtime. IOT LTE modems made by Digi, for example, include extra non-volatile and volatile storage and a MicroPython runtime so that business logic can run directly in the modem. You can bet that there are Hayes commands involved.

So 40 years later, a huge variety of modern electronics using cutting-edge cellular data networks are still, at least for initial setup, pretending to be a 300-baud Hayes Smartmodem. Maybe you can still find a case out there where coercing another computer to attempt to send “+++” followed by, after a sufficient pause, “ATH” will cause it to drop off the network.

A final tangent on Hayes commands, and what brought them to my mind: through a combination of good luck and force of will I have managed to get a dealership to take my money for a new car (this proved astoundingly difficult). Since Albuquerque is quite competitive in its effort to regain the recently lost title of Car Theft Capitol of the USA I have fit it with a tracking device. This device, made by a small Chinese outfit, runs the entirety of its logic within a modem with extended firmware. This is sometimes called a “microcontroller-less” design, although obviously the modem is essentially functioning as a microcontroller in this case. For configuration, the tracker exposes the modem’s Hayes serial interface on an external connector, and the vendor provides a software tool that generates very long Hayes command strings to configure the tracker behavior (endpoint, report frequency, immobilize logic, etc). It’s possible to use AT commands on this interface to send and receive SMS, for example, which makes the tracker a more flexible device than it advertises.

Actually, I lied, one more tangent: Wikipedia notes that the Smartmodem used a novel design of an extruded aluminum section that the PCB slid into, and a plastic cap on each end. This was an extremely common case design for ’90s computer accessories. Cheaper plastic injection molding seems to have mostly killed it off, but it was super convenient to take these types of cases apart and I rather miss them now.

[1] In fact a new and somewhat upcoming GSM feature called “eCall” enables “data-only” emergency calls, mostly intended for use by in-vehicle assistive technologies that may connect an emergency call and then send a position and status report under the assumption that the occupants may be incapacitated and unable to speak.

[2] Note that newer modems and operating systems are starting to use MBIM more often, a newer USB profile that includes a newer command channel. If you have an LTE modem and do not see the expected Hayes serial device, MBIM may be the reason... but on most modems a Hayes command must be issued to switch the modem to MBIM mode, so the Hayes command set is still in use even if only briefly.