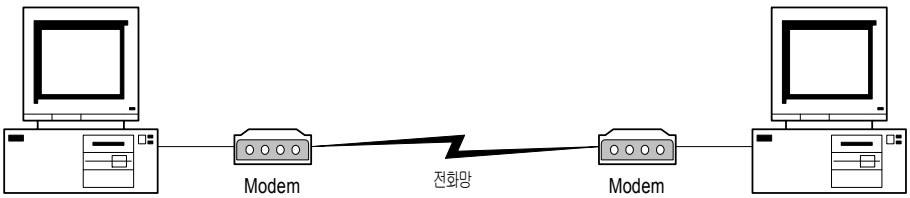
4장 모뎀

현재, 전 세계에 걸쳐 대부분의 전화 연결은 POTS(Plain Old Telephone Service)형식이다. 이 POTS는 전화기와 전화국의 교환기 사이의 연결 부분을 제외하고는 디지틀로 처리된다. 즉, 아직까지 전화기와 전화국 교환기 사이에는 사람 목소리가 아날로그 방식으로 전달된다는 말이다. 이 때문에 컴퓨터에서 나오는 디지틀 데이터를 전화선을 통해 전달하려면 부득이 아날로그 형태로 변환해 주어야 한다. 또한 전화선을 통해 전달되는 디지틀 데이터를 컴퓨터에서 받으려면 전달되는 아날로그 데이터를 다시 디지틀 데이터로 바꿔 주어야 한다. 디지틀에서 아날로그로(변조:MOdulation), 이를 다시 아날로그에서 디지틀(복조:DEModulation)로 바꿔주는 장치가 바로 모뎀이다.

모뎀(MODEM)은 변조(MOdulation)와 복조(DEModulation)의 합성어로서 우리말로 풀이하면 변조-복조기 또는 변복조기 정도가 되겠다.



RS-232가 처음 나왔을 때, 모뎀은 동기식 반이중(half-duplex) 장치였으며, 저속에서는 꽤 잘 동작했다. 반이중이란, 송수신 작업에 동시에 이루어 질 수는 없고, 송신이면 송신, 수신이면 수신, 한번에 하나씩만 이루어지는 것을 말한다. 반이중통신에서는 RTS(Requsest To Send) 신호를 통해 전송 허가를 요구해야 했으며, CTS(Clear To Send)를 통해 허가가 나기를 기다려야 했다. 전송이 허락되면 접속된 상대방은 이 전송작업이 끝날 때 까지 다른 일은 할 수 없다.

현재의 모뎀은 전이중(full-duplex) 방식이며 속도은 최소한 9600bps이상이다. 현재는 28,800bps모뎀이 일반적으로 사용되고 있으며, 압축등의 방법을 이용하면 최대 57,600bps정도 속도의 모뎀까지 가능할 것이다.

역사적으로 모뎀은 크게 두가지 면에서 발전해 왔다. 하나는 속도이고, 다른 하나는 지능이다. 1970년대 초반의 초기 모뎀은 110~300bps였지만 오늘날은 28,800bps가 대중적이며, 57,600bps도 곧 선보인다고 한다. 초기의 300bps모뎀은 말그대로 전압을 주파수 출력으로 바꿔주는 역할만 했지만, 현재의 모뎀은 매우 지능적으로 되어서 컴퓨터와 논리적으로 상호작용을 할 뿐만 아니라, 오류 정정과 전송시 자료 압축등을 수행하기도 한다.

모뎀 속도

Bell 103

최초의 표준 비동기 모뎀으로 속도가 110~300bps까지 가능했으며 전이중 통신방식이 지원되었다.

Bell 212, V.22

속도가 네배까지 증가한 1,200bps까지 가능. 212형의 대부분의 모뎀은 이전의 103 모델과도 연결할 수 있었다. 이러한 호환성(새로 나오는 제품이 이전에 나왔던 제품의 기능을 모두 포함)은 이후에도 일반적인 경향이 되었다.

V.22bis

이 모뎀에 와서, 비로소 하나의 표준으로 정착되게 되었는데 이 모뎀은 2,400bps 속도까지 지원되었다. 80년대 후반까지 가장 주류를 이루었던 모뎀이 바로 2,400bps 모뎀이었다. 필자도 바로 이 2,400bps모뎀으로 처음 통신을 했던 기억이 난다. 당시의 KETEL등의 PC통신 서비스들은 거의 대부분 2,400bps를 지원했던 것으로 기억한다.

V.32

9,600bps모뎀이 표준이 되기까지는 상당한 시간이 걸렸다.

V.32bis

14,400bps속도까지 가능. V.32와 호환. 느린 통신속도에 대한 불만이 커져갔기 때문에 이 표준은 빠르게 정착되었다. 모뎀 제작자들은 V.32모뎀을 만드는 동안 많은 것을 배웠기 때문에 V.32bis모뎀을 내 놓을 때는 좀 더 싼 가격에 만들어 낼 수가 있었다.

이 때문에 14,400bps모뎀은 통신 프로그램을 위한 새로운 표준으로 쉽게 정착되었다.

좀 더 빠른 속도를 지원하는 새로운 표준이 속속 발표되고 있으며 전화선로 상에서 통신속도의 한계라고 하는 57,600bps까지는 지원될 것이다.

모뎀 지능

벙어리(dumb) 모뎀

초창기 110~300bps급 모뎀을 부르는 말로서, 모뎀 내에 마이크로 프로세서가 없어서 전화를 건다거나, 수화기를 든다거나(off-hook), 수화기를 내려놓는다거나(on-hook)하는 등등의 작업을 할 수 없었던 모뎀을 말한다.

스마트(Smart) 모뎀

똑똑한, 지능있는(smart) 모뎀은 마이크로프로세서가 내장된 모뎀을 말하는 것으로 흔히 스마트 모뎀이라고 하면 헤이즈사의 스마트모뎀(Hayes Smartmodem)을 일컫는 것이다. 헤이즈사에서는 처음에 300bps에 이 지능을 채용했다가 이후에 1,200bps과 2,400bps에도 채용했다. 스마트 모뎀은 RS-232를 통해 컴퓨터와 통신을 했는데 초기에는 전화걸기 정도의 기능밖에 없었다. 이제는 표준이 되어버린 "AT"명령어 집합을 이용해서 "ATD555-2555"라고 입력하면 전화는 off-hook이 된 다음 555-2555로 연결을 시도하는 다이얼링을 하게 된다. 우리가 흔히 알고 있는 AT명령은 바로 이 헤이지사에서 개발된 것이다.

MNP-4 모뎀

마이크로콤(Microcom, Inc.)이란 회사는 MNP라고 하는 프로토콜을 개발했다. MNP, MNP-1, MNP-2, MNP-3은 별로 개선된 점이 없었다. 동기화 프로토콜을 사용해서 자료를 전송함으로써 시작비트와 정지비트를 제거해서 20%정도의 속도를 증가시켰다.

MNP-4에 와서는 크게 개선된 점이 있었는데, MNP-4는 오류 검출-정정 프로토콜로

서 두 개의 DTE(Data Terminal Equipment: PC통신에서는 그냥 PC라고 생각하면 된다)간의 연결에서 오류가 전혀 없도록 해주는 획기적인 것이었다. MNP-4를 사용하는 모뎀끼리 연결되면 전달, 수신 되는 정보에는 부가적인 정보가 추가되며 오류가 발생하면 오류가 발생한 블록을 재전송하게 된다.

MNP-4급의 모뎀이 사용되기 시작하면서 응용프로그램 개발자들은 데이터가 제대로 전송되는지 일일이 걱정하지 않아도 되게 되었다.

MNP-5 모뎀

MNP-5는 자료를 전송하기 전에 이를 압축한다. MNP-5는 적응 허프만 코딩(adaptive Huffman coding)을 사용해서 크게는 1/2까지 압축한다. 따라서 모뎀의 실질적인 속도는 2배정도 향상되는 것이다. MNP-5는 즉시 성공했지만 한가지 결점이 있었다. 이미 다른 방법으로 압축되어 있는 데이터의 경우에는 거의 압축이 되지 않기 때문이었다. MNP는 독점적인 프로토콜이었기 때문에 이 프로토콜을 자신의 모뎀에 적용하려면 마이크로콤사의 허락을 얻어야 했다. 마이크로콤사는 이 프로토콜을 표준화함으로써 많은 이익을 보았다.

V.42, V.42bis

MNP-4, MNP-5는 비동기 모뎀에서 오류정정과 자료압축의 선구자적인 역할을 했다. 이 마이크로콤사의 프로토콜은 몇가지 기능적인 개선을 해서 CCITT(현 ITU) 표준으로 자리잡았으며, 그것이 바로 V.42와 V.42bis이다.

V.42는 오류검출과 오류가 생긴 자료를 다시 전송하기 위해 LAPM(Link Access Protocol for Modems)을 사용한다.

V.42bis는 자료압축에 대한 표준이다. 이 표준은 LZW기반의 자료압축 알고리듬을 사용하며 MNP-5보다 더 강력하다. 압축이 크게는 1/4정도까지 된다.

악수: 핸드쉐이킹(handshaking)

대부분의 고속 모뎀은 모뎀이 장착된 PC와의 통신에 "locked-DTE"라는 방법을 사용

한다. "locked-DTE"란 다른 곳에 어떤 속도로 연결되든 장착된 PC와 모뎀 사이의 연결은 항상 같은 보 레이트(baud rate)를 유지하는 것을 말한다. 예를 들어, V.32모뎀은 V.22bis모뎀과 연결될 때는 2,400bps이지만, 자신이 장착된 PC와는 38,400bps를 유지한다. 컴퓨터에서 모뎀으로는 38,400bps의 속도로 자료가 전송되지만, 전화선을 통해서는 2,400bps로밖에 전송되지 않는다는 말이다. 이렇게 되면 이 둘의 속도 차이 때문에 명백히 문제가 발생한다. 컴퓨터에서는 38,400bps의 속도로 계속 자료를 쏟아내고 모뎀은 이를 모두 처리하지 못하기 때문에 모뎀의 내부 버퍼는 금방 가득 차 버릴 것이다. 이것을 막으려면 "악수"하는 수밖에 없다. 즉, 보내는 쪽이 신사적으로 "이제 보내도 되겠소"하면 받는 쪽에서 "좋소"하면서 서로 "악수"하는 방식을 말한다. 이렇게 자료의 흐름에 대한 제어를 흐름 제어(flow control)라고 하며 흐름 제어 방식에는 여러 가지가 있다. 비동기 RS-232연결에서 사용되는 대표적인 흐름제어 방식으로 RTS/CTS와 XON/XOFF가 있다.

하드웨어 흐름 제어(hardware flow control)

RTS/CTS 흐름제어는 하드웨어 흐름제어 방식의 일부이다. 하드웨어 흐름제어는 직렬 데이터 흐름의 시작과 멈춤을 RS-232 제어선을 이용해서 수행한다. 컴퓨터같은 DTE장비는 RTS를 모뎀에서 나오는 데이터의 시작과 멈춤을 제어하는데 사용하고, 모뎀같은 DCE장비는 CTS를 컴퓨터에서 나오는 데이터의 시작과 멈춤을 제어하는데 사용한다.

이론적으로, RTS/CTS 흐름 제어는 구현하기가 비교적 단순하다. CTS사용의 예를 들어 보자. 모뎀의 입력 버퍼는 어떤 일정한 크기로 제한되어 있는데, 새로운 문자를 읽어 들일 때마다 모뎀은 매번 버퍼가 가득찼는지 검사한다. 버퍼가 일정 수준 이상 차면, CTS 선이 off로 떨어진다. DTE는 이 선이 off되면 전송을 중단하게 된다. 모뎀의 수신 버퍼가 수신된 데이터를 처리해서 버퍼의 데이타가 일정 수준 이하로 줄어들게 되면, 다시 모뎀은 CTS를 on으로 만들고, DTE는 다시 전송을 시작하게 된다.

반대로 DTE, 즉 컴퓨터 쪽의 입력 버퍼도 일정 크기로 제한되어 있으며, 이 버퍼가 일정 수준 이상으로 차면 RTS가 off되면 일정 수준 이하로 내려가면 RTS는 on된다. 모뎀은 이 RTS의 on/off 상태를 보고 데이터를 DTE쪽으로 전송할 것인지 기다릴 것인지를 결정한다.

RTS/CTS가 가장 일반적인 하드웨어 흐름제어 방식이긴 하지만, DTR/DSR도 자주 사용된다.

소프트웨어 흐름 제어(software flow control)

하드웨어 흐름제어는 별로 추천할만한 것이 못된다. 왜냐하면 하드웨어 흐름제어는 RTS, CTS, DTR, DSR같은 특정한 하드웨어 선이 필요하기 때문이다. 소프트웨어 흐름 제어는 전통적으로 XON/XOFF 프로토콜이 사용되어 왔다. XON/XOFF 프로토콜은 다음과 같은 방식으로 동작한다. 데이터를 수신하는 쪽에서 받을 수 있는 최대 수준을 미리 정해 놓고, 데이터를 받는다. 받은 데이터 양이 일정 수준 이상으로 넘어가면 XOFF(Control-S, ASCII 19)를 데이터를 보내는 쪽으로 보낸다. 데이터를 보내는 쪽에서는 이 XOFF를 받으면 데이터 보내기를 중지한다. 이제 다시 데이터를 받는 쪽에서 받은 데이터가 어느 정도 처리되어 일정 수준 이하로 내려가면, 다시 보내는 쪽에 XON(Control-Q, ASCII 17)을 보낸다. 보내는 쪽에서는 이 문자를 받으면 보내기를 다시 계속하게 된다.

XON/XOFF 문자는 원하는 경우 다른 문자를 사용할 수도 있다.

이 XON/XOFF의 결정적인 문제점은 이진(binary) 데이터를 보낼 때 생긴다. 데이터 중에 XON/XOFF 문자가 들어 있을 수 있기 때문이다.

모뎀 명령어

모뎀에 대한 몇가지 표준이 있긴 하지만, 모뎀 명령어에 있어서는 어느 정도 표준화되어 있어서 미국의 헤이즈 마이크로컴퓨터 프로덕트에서 개발한 AT 또는 헤이즈 명령어가 표준 모뎀 명령어라고 할 수 있다. 모든 AT명령어들은 동일한 기본구조를 가지고 있다. 모든 명령어는 AT로 시작하는데, AT는 ATTENTION의 앞 두자로 생각된다. 이 뒤에 여러가지 문자로 구성된 명령이 있고 마지막에는 캐리지 리턴(ASCII 0x0D) 문자가 들어간다. 모뎀 명령어는 대소문자 구분을 하지 않는다.

모뎀에는 두개의 운영상태가 있다. 명령어 상태와 온라인 상태인데 먼저 명령어 상태에서는 모뎀의 운영을 제어하기 위해 AT명령을 내릴 수 있게 되어 있다. 온라인 상태에서는 모뎀으로 보내진 모든 데이타가 상대방 시스템으로 보내지며 모뎀의 AT명령어도 데이타의 일부로 생각해 버린다. 온라인 상태에서 모뎀이 인식하는 유일한 명령어는 이스케이프 명령어 뿐이다. 이스케이프 코드는 일반적으로 '+'(ASCII 43) 문자인데, 다른 문자로 바꿀 수도 있으며, 이 값은 S-레지스터에 저장된다. S-레지스터는 아래 부분에서 설명한다. 모뎀에서 ATD명령으로 전화를 걸어 연결이 되면, 모뎀은 온라인 상태로 들어가게 되는데, 이 온라인 상태에서는 사용자가 입력한 키 입력이 그냥 연결된 상대방 시스템으로 전달될 뿐 모뎀에 직접 명령을 내릴 수는 없다. 그렇다면, 전화 끊기 명령인 ATH는 어떻게 내릴 수 있을까? 온라인 상태에서 AT명령을 내리려

면 잠시 온라인-명령상태로 전환해야 하는데, 이 전환은 바로 이스케이프 코드를 일정 시간 간격으로 연속적으로 보내주는 것이다. 이 시간 간격 역시 S-레지스터에 저장된다. 일반적인 이스케이프 코드인 '+'를 일정간격으로 연속 세 번 보내면 모뎀은 "OK\r\n"응답을 보내며 전화는 끊어지지 않은 상태로 잠시 온라인 명령 상태로 전환되게 된다. 이 때 ATH명령을 내려주면 이 AT명령은 모뎀 명령으로 인식되고, 따라서 전화가 끊기는 것이다.

S-레지스터

모뎀에는 S-레지스터라는 것이 있는데 여기에는 모뎀의 상태에 대한 전반적인 정보가 저장되어 있다. 필요에 따라 이 값을 읽어 오거나 새로운 값을 써넣을 수도 있다. 이 레지스터는 각 모뎀 칩 제조업체마다 조금씩 다르기 때문에 여기서는 헤이즈 사의 스마트 모뎀의 S-레지스터 중 S0~S12까지만 살펴보도록 하자. 이 값들은 직접 AT명령으로 바꿀 수 있는 것이 있고, 바꿀 수 없는 것이 있다. S0는 직접 바꿀 수 있는 대표적인 것이다.

S0 - Number of Rings to Auto answer

전화 받기까지 벨 수를 지정한다. 즉, ATS0=n하면 전화벨이 n번 울리면 모뎀은 전화를 자동으로 받게 된다. n=0이면 전화를 받지 않으며, 이 값이 모뎀의 기본값이다. 따라서 보통 상태에서는 모뎀은 전화벨이 울려도 전화를 받지 않는다. 친구에게 전화를 걸어 자기 집으로 이야기 같은 통신 에뮬레이터를 이용해서 전화를 걸어보라고 해보라. 그리고 자신은 통신 에뮬레이터에 ATS0=1이라고 해 두면, 전화벨이 울리자 마자 모뎀이 전화를 받게 된다. 이 상태에서 자신이 화면에 입력한 것이 상대방 화면에 그대로 보이게 된다. 이것이 바로 모뎀을 이용한 일대일 통신이다.

범위: 0~255

기본값: 0

S1 - Ring Counter

현재 벨 수를 저장해 놓는다. 현재까지 벨이 몇번 울렸는지를 알려준다. 벨이 8초동안 더 이상 울리지 않으면 이 값은 0으로 다시 초기화 된다.

범위: 0~255

기본값: 0

S2 – Escape Character

온라인 통신 상태에서 온라인 명령 상태로 전환하는데 사용되는 문자.

범위: 0~255

기본값: '+' : 43

S3 – Carrage Return Character

명령행과 결과코드의 끝문자.

범위: 0~127

기본값: 13

S4 – Line Feed Character

line feed 문자

범위: 0~127

기본값: 10

S5 – Backspace Character

백 스페이스 문자

범위: 0~127

기본값: 8

S6 – Wait Time for Dial Tone Before Blind Dialing

발신음(dial tone)이 검출 될 때까지 기다리는 시간을 지정. 발신음이란 수화기를 들었을 때 '뚜--'하는 소리를 말하는 것으로 전화를 걸 수 있는 상태가 되었다는 뜻이다. 전화걸기 명령인 ATD에서 W를 붙여주면 발신음이 검출될 때까지 전화를 걸지 않는데, 이 때 이 발신음 검출 대기 시간을 말하는 것이다. 기본값은 2로 2초 내에 발신음이 검출되지 않으면 전화걸기에 실패한다. 일반적으로 가정에서는 수화기를 들자마자 발신음이 검출되지만, 구내 교환기가 설치된 사무실의 경우에는 수화기를 들면 '뚜-뚜-'하는 소리가 들리고 '9'를 눌러야 비로소 '뚜--"하는 소리가 들린다. 따라서 구내 교환기가 설치된 곳에서는 9를 누르기 전에는 발신음이 검출되지 않으므로 보통은 발신음은 무시하고 그냥 전화걸기(dialing)를 하는 것이 좋다.

범위: 2~255초

기본값: 2초

S7 – Wait Time For Carrier After Dial

전화를 건 후 반송파가 검출될 때까지 대기 시간

범위: 1~255초

기본값: 30

S8 – Pause Time For Dial Delay

ATD명령에서 ','는 다이얼링을 잠시 멈추라는 명령이다. 즉, "ATDT9,555-2555\r"하면 9번을 누른 후 잠시 후에 555-2555를 다이얼링 한다. 이렇게 하면 구내 교환기가 있는 곳에서 다이얼 톤이 뜨는 시간까지 약간의 시간 지연을 줌으로써 다이얼링이 잘 된다. ','의 시간 지연 값. 기본 값은 2초이다.

범위: 0~255초

기본값: 2

S9 – Carrier Detect Response Time

반송파 검출 시간

범위: 1~255

기본값: 6(0.6초)

S10 - Lost Carrier To Hangup Delay

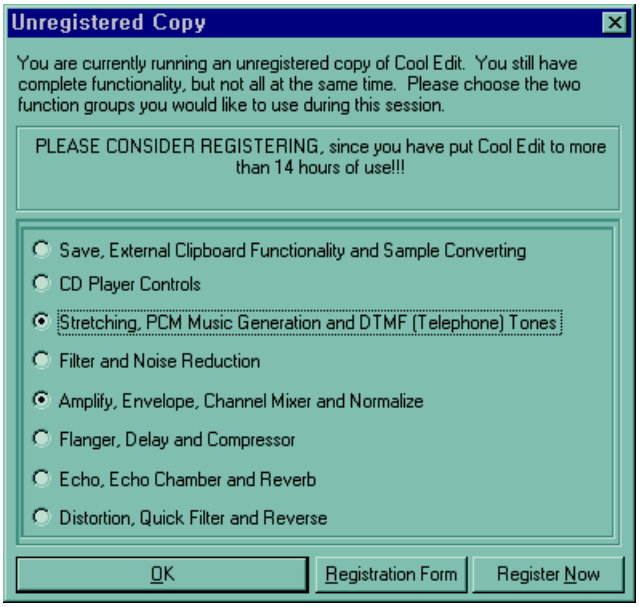
무신호시 차단 시간

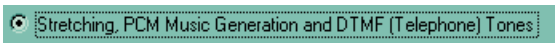
범위: 1~255초

기본값: 7(0.7초)

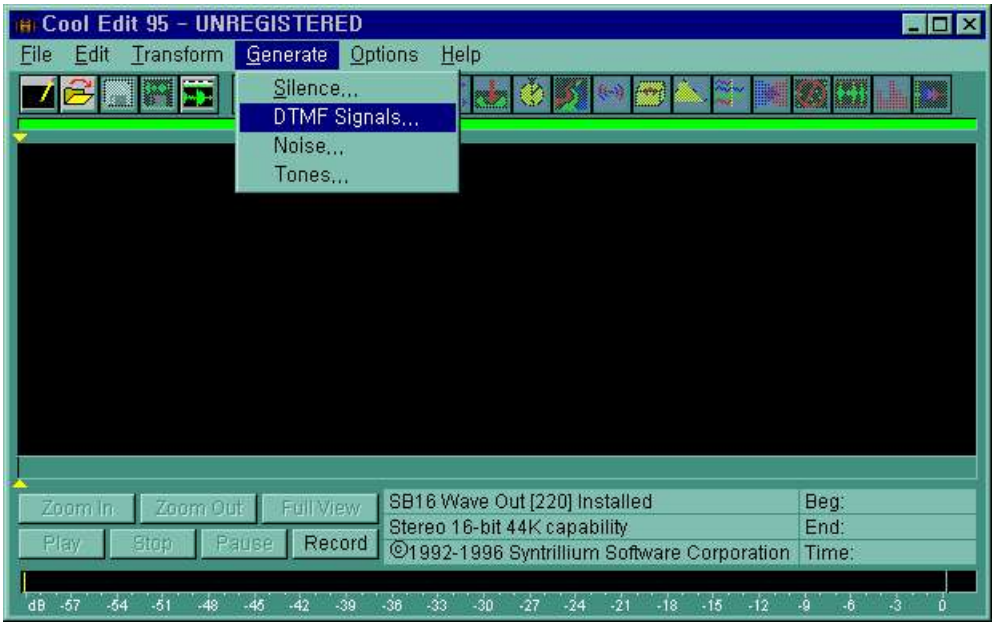
S11 - DTMF Tone Duration

옛날 기계식 전화기는 다이얼을 돌리는 방식이었다. 다이얼을 돌리면 '뚜루루루-' , '뚜루루루-'하는 소리가 들리는데 이 소리가 바로 전화국의 교환기로 전달되어 숫자 몇 번을 다이얼링했는지를 알려주는 것이다. 하지만 요즘은 이런 전화기는 보기 힘들어졌다. 지금 사용하는 전화기는 버튼을 누르면 '뚜-'소리가 나는 전자식 전화기이다. 전화기에 있는 각 숫자를 눌러보면 각기 다른 소리가 나는데, 이 소리가 전화국 교환기로 전달되어 다이얼링이 되는 것이다. 한가지 재미있는 실험을 해보자. 스피커 폰 기능이 있는 전화기를 이용해서 전화기 버튼을 누를 때 나는 소리를 녹음해 두고, 이 소리를 수화기를 들고, 말하는 쪽에다가 대고 재생을 해보라. 깨끗하게 녹음이 되었다면 전화가 걸린다. 거짓말 같지만 정말이다. 녹음하기 어렵다면 다음과 같은 방법으로 확인해 보라. CoolEdit이란 소리 파일 편집 프로그램을 이용하자.

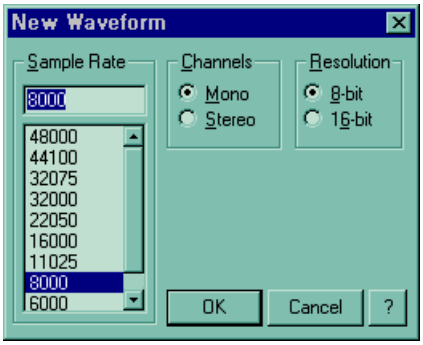


처음 시작할 때 

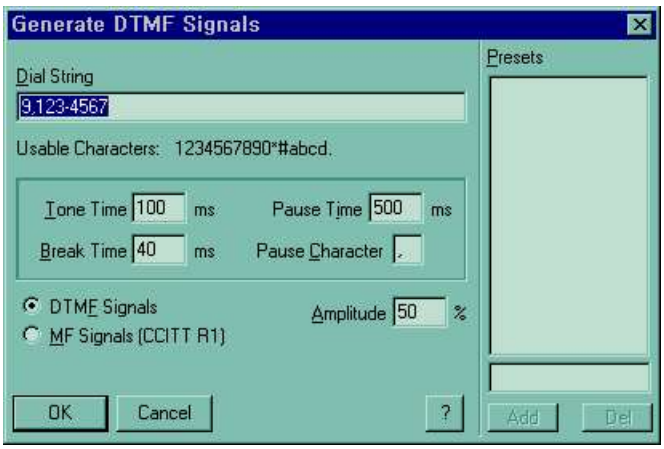
을 선택한다.



Generate-DTMF Signals...를 선택한다.



기본값을 선택하고,



자신이 원하는 전화번호를 적는다. 그런다음 OK를 누르면

위의 그림과 같이 소리가 생성된다. 이제 수화기를 들어, 컴퓨터 스피커에 갖다대고

Play를 눌러보라. 전화가 걸리는 걸 확인할 수 있을 것이다. 이처럼, 전화 다이얼링은 단순히 해당 주파수의 소리를 수화기를 통해 보내면 되는 것이다.

DTMF Tone이란 전자식 전화기의 숫자나 '*', '#'버튼을 눌렀을 때 나는 소리이다. 사람이 전화를 걸 때는 버튼을 하나씩 눌러서 걸지만, 모뎀은 ATD명령으로 전화번호를 한꺼번에 받아서 전화를 건다. 전자식 전화기의 경우에는 "ATDT555-2555\r"라고 입력하게 되는데 "555-2555"를 하나씩 다이얼링 할 때 각 숫자 사이의 시간 간격이 바로 S11에 저장된다. 기본값은 0.7초로 더 짧게 하면 더 빨리 다이얼링 된다. 그렇다고 너무 짧게 하면 연결이 잘 되지 않는다. 예를 들어, ATS11=50하면 그 간격이 0.5초로 된다.

범위: 50~255밀리초

기본값: 70(밀리초)

S12 - Escape Prompt Delay

세 개의 이스케이프 코드 문자의 입력 시간 간격. 온라인 상태에서 이 시간 간격으로 세 개의 이스케이프 문자가 입력되면 온라인 명령상태로 잠시 빠지게 된다..

범위: 0~255

기본값: 50(1초)

모뎀 명령어(AT COMMAND SET)

대부분의 모뎀 칩들은 헤이즈사의 스마트모뎀에 채택된 모뎀 명령어를 표준처럼 사용하고 있다. 모뎀 칩 제조업체마다 약간씩 추가된 명령이 있긴 하지만 기본적인 명령은 모두 같으므로 별로 신경쓰지 않아도 된다.

모뎀 명령어는 대소문자 구별없는 영문자를 사용하며 항상 AT(ATtention이란 의미라고 생각된다)로 시작된다. 바로 이전에 내린 명령을 반복하라는 명령인 'A/'는 예외로 'AT'없이 그냥 'A/'를 입력한다.

A/ - 최종 명령 반복

가장 최근에 내린 명령을 반복한다. 즉, 이전에 'ATDT555-2555'로 전화를 걸었는데 통화중인 경우에 다시 'ATDT555-2555'할 필요없이 그냥 'A/'하면 'ATDT555-2555'한 것과 같다는 말이다.

ATA - 전화 받기

전화에 응답하기. 이야기 같은 통신 에뮬레이터를 띄워둔 상태에서 전화가 오면 화면에는 'RING'이 벨이 울릴 때마다 표시된다. 이때 'ATA'를 입력하면 전화를 받게된다. S0레지스터에 벨 수를 지정해 주면, 지정된 벨 수만큼 울린 다음 전화를 받는다. 그 전에라도 전화받는 쪽에서 'ATA'라고 치면 즉시 전화를 받게 된다.

또 하나 재미있는 것이 있는데, 1대 1 통신을 하고자 할 때, 보통은 다음과 같은 방식으로 하는 사람이 대부분이다. 한사람이 전화를 걸어 "내가 모뎀을 이용해서 전화를 걸 테니, 넌 이야기 같은 통신 에뮬레이터를 띄어서 1:1통신 메뉴를 선택하든지, 아니면 직접 'ATS0=1'이라고 입력하고 내 전화를 기다려랴" 라고 한 다음, 컴퓨터 앞에 가서 통신 에뮬레이터를 이용해 전화를 건다. 하지만, 이 방법은 좀 문제가 있다. 전화도 두 번 걸어야 하고, 통신 에뮬레이터로 전화 거는 사이에 1:1통신 대기 중인 집에 다른 사람에게서 일반 전화가 걸려오면 그 집에 전화 건 사람은 '삐~~삐리리릭~'하는 이상한 소리만 듣게 될 것이다.

다음과 같은 방법을 한번 써보라. 먼저, 그냥 전화를 한다. "1:1 통신을 할 테니 PC에 가서 통신 에뮬레이터를 띄워라"고 한 다음, 한쪽에서는 'ATD'를 입력한다. 그러면 다른 한쪽에서는 '삐~~삐리리~~'하는 소리가 들릴 것이다. 이 때 재빨리 'ATA'를 입력하면 이제 서로 1:1통신이 가능한 상태가 된다. 음성통화 중에 데이터 통신 상태로 전환된 것이다. 화면에 글자를 입력하면 상대방 화면에 입력한 글자들이 표시된다. 이 상태에서는 서로의 약속하에 파일을 보내고 받는 것도 가능하다. 방법은 PC통신 서비스 사용하는 것과 별반 다르지 않다. 파일을 보내고자 하는 쪽에서는 파일 올리기(UPLOAD)를 시도하고, 파일 받는 쪽에서는 파일 내려받기(DOWNLOAD)를 시도하면 된다. 시간 차이가 아주 많이 나지 않는 한 성공할 것이다.

ATBn - 표준 선택

CCITT V.22와 BELL 212A/103 표준 중 하나를 선택하는 명령이다. n=0이면 CCITT V.22이고 n=1이면 BELL 212A/103이다. 속도가 300bps, 1200bps일 경우에 사용하던 것으로, 사실상 지금은 무시해도 좋다. 왜냐하면 다른 속도에서는 모두 CCITT 표준을 따르기 때문이다.

ATDn – 전화 걸기

우리가 알고 있는 'ATDT'나 'ATDP'말고도 'ATD'뒤에 올 수 있는 문자가 몇가지 더 있다. 뒤에 올 수 있는 문자들을 정리해 보면 다음과 같다.

문자	기능
T	전자식 전화기 방식으로 전화를 건다. 톤(tone)방식이라고도 한다. 전화를 걸면, 전자식 전화기처럼 소리가 난다.
P	기계식 전화기 방식으로 전화를 건다. 펄스(pulse)방식이라고도 한다. 전화를 걸면, 옛날 전화기처럼 다이얼 돌아가는 소리가 난다.
W	전화 걸기 전에 전화선에서 발신음(dial tone)이 들리는지 확인한다. 발신음이 들리지 않으면 전화를 걸지 않는다. 발신음이란 수화기를 들었을 때 나는 '뚜--'하는 소리를 말한다.
,	일시 정지. S8레지스터에 저장된 시간만큼 일시 정지한다. 구내 교환기가 있는 경우, 9를 눌러야 발신음이 들리기 시작한다. 이 경우 'ATDT9,555-2555'하는 식으로 명령을 내리면 일단 9가 눌린 다음 S8에 저장된 시간만큼 나머지 '555-2555'를 다이얼 하지 않고 기다린다.
;	전화걸기를 한 다음, 온라인 상태로 들어가지 않고 계속해서 명령상태로 남아있게 한다. 온라인 상태로 들어가려면 'ATO'명령을 이용하면 된다. ';'를 추가하지 않고 그냥 전화를 거는 것이 일반적인데, 이렇게 하면 모뎀은 명령상태에 있다가 전화걸기를 하고 나서 온라인 상태로 들어가게 된다. 온라인 상태에서는 입력한 문자를 모뎀 명령으로 인식하는 것이 아니라 그냥 입력된 데이터라고 생각한다. 따라서 온라인 상태에선 어떤 모뎀 명령도 인식되지 않는다. 모뎀 명령을 내리려면 온라인 상태에서 전화를 끊지 않는 채로 잠시 온라인 명령상태로 나와야 하는데, 이 때 사용하는 명령은 S2레지스터에 저장된 문자(보통은 '+', ASCII 43)를 일정간격으로(S12레지스터에 지정) 으로 보내주는 것이다. 이렇게 하면 'OK'응답과 함께 온라인 명령 상태로 상태가 바뀐다. 이 상태에서는 모뎀 명령이 입력된다. 따라서 통신 중에 전화를 끊으려면, '+'를 일정간격으로 세 번 모뎀에 입력한 다음 'OK'응답을 받은 후에 'ATH'를 입력하면 된다.

전화번호 중에 '-'나 '(', ')'같은 문자들은 모두 무시되므로 'ATDT(02)-555-2555'나 'ATDT0255552555'나 똑같이 동작한다.

ATEn – 자국 반향

모뎀이 사용자가 입력한 문자를 다시 반향(echo)할 지 하지 않을 지 설정한다. 보통 '자국반향'이라고 부른다. 모뎀으로 한 문자를 보내면, 모뎀이 자신이 받은 문자를 다시 직렬포트의 입력버퍼에 넣어 줄 지 주지 않을 지를 결정한다. 따라서 'ATE0'하고 엔터를 쳐서 'OK'응답을 받고 나면 그 다음부터는 명령어를 입력해도 화면에는 아무것도 나타나지 않는다. 이럴 때, 'AT'하고 엔터를 치면 'AT'는 나타나지 않지만 'OK'라는 응답은 제대로 나타난다.

일반적인 경우에는 반향이 있는 것이 편하다. 그렇다면 반향을 꺼야하는 경우는 어떤 때인가? 모뎀에 어떤 명령을 내린 다음, 그 응답을 읽어 분석하는 것을 생각해 보자. 대표적인 예로 전화걸기 대화상자 같은 경우가 있다. 전화걸기 대화상자에서 어떤 항목을 선택해서 전화를 거는 경우에, 전화가 연결되었는지, 통화중인지, 전화선에 이상이 있는지는 모뎀이 돌려주는 결과값으로 알 수 있다. 'ATDT01410\r'라는 문자열을 직렬포트에 쓰면, 모뎀은 이 명령에 따라 01410번으로 톤방식('T')을 이용해 전화를 건다. 모뎀 명령어는 '\r'로 끝난다. 모뎀에 명령을 내리려면 문자열 끝에 꼭 '\r'(캐리지 리턴)을 붙여주어야 한다. 전화가 연결되면 'CONNECT xxxx'라는 응답이 오고, 회선에 이상이 있거나 하면 'NO CARRIER'라는 응답이 온다. 이 응답을 분석해야 하는데, 만약 자국반향이 켜져 있다면, 직렬포트의 출력 버퍼에는 'ATDT01410'\r'CONNECT 9600\r\n'가 들어오게 된다. 우리가 분석해야 하는 응답은 다이얼링 후의 'CONNECT xxxx\r\n'인데, 입력된 다이얼 명령인 'ATDT01410\r'까지 버퍼에 들어있게 된다. 이 때문에 응답 분석이 아주 귀찮게 되 버렸다. 이 때, 자국반향을 꺼버린 상태에서 명령을 내리고, 응답을 받으면, 훨씬 간단하게 될 수 있다. 통신에뮬레이터 같은 프로그램상에서는 자국반향을 켜는게 기본이지만, 내부적으로 전화걸기 등을 할 때는 잠시 자국반향을 꺼 두고 응답을 분석하는 것이 더 수월하다.

ATE0 : 입력된 문자가 반향되지 않음

ATE1 : 입력된 문자가 반향됨. 기본값.

ATHn – 전화 끊기

전화를 끊는다. 보통은 'ATH'만 입력해도 된다. 전화 끊기 말이 나왔으니, on-hook과 off-hook에 대해서 정리하고 넘어가자. 전화기 수화기 밑에 있는 스위치를 훅(hook) 스위치라고 하는데, 바로 이걸 떠 올리면 on-hook과 off-hook을 쉽게 구분할 수 있다. 수화기가 그대로 놓여 있다면 훅 스위치가 눌린 상태, 즉 on-hook이고, 수화기를 들면, 훅 스위치가 떨어진 상태, 즉 off-hook상태이다. 그러므로, on-hook이란 전화가 끊긴

상태를 말하며, off-hook이란 수화기를 든 상태, 즉 '뚜--'하는 소리가 들리는 상태 쯤으로 생각하면 된다.

ATH0 : 전화를 끊는다.

ATH1 : on-hook상태라면 off-hook상태로 만든다. 모델에 따라 off-hook상태로 그대로 있는 것이 있고, 일정시간 후에 on-hook상태로 돌아가는 것이 있다.

ATIn - 모뎀 확인

모뎀 제품의 구성등을 확인 할 때 사용한다. n값은 모뎀 제작회사 마다 차이가 있을 수 있다. 일반적으로는 모뎀 칩의 제품 코드, ROM 체크섬 값, 제품 명등의 아스키값 등등을 확인할 수 있다.

ATLn - 스피커 소리크기

모뎀에는 소형 스피커가 부착되어 있는데, PC통신 서비스 업체에 접속하고자 할 때, 나는 소리는 바로 이 소형 스피커에서 나는 소리이다. 또한 전화를 잘못 걸어 상대방에서 사람이 전화를 받았을 때, 모뎀에서 사람 목소리가 그대로 나오는 것을 들을 수 있는데, 바로 이 소리가 모뎀 스피커를 통해 흘러 나오는 것이다.

ATL0, ATL1 : 작게(기본값)

ATL2 : 중간

ATL3 : 크게

ATMn - 스피커 제어

상황에 따라 스피커를 켜거나 끈다.

ATM0 : 항상 스피커를 끈다. 이 때는 모뎀에서 어떤 일이 일어나는지 전혀 들을 수가 없기 때문에 별로 사용되지 않는다. 다만, 자신이 집에서 사설 BBS등을 운영하고자 할 경우에, 계속되는 모뎀 접속하는 소리 때문에 방해를 받고 싶지 않다면, 이 명령을

사용해서 조용한 상태에서 BBS를 운영해 나갈 수 있다.

ATM1 : 접속이 진행되는 동안만 켜두고, 접속이 완료되면 스피커를 끈다.

ATM2 : 스피커를 항상 켜둔다.

ATM3 : 접속이 되었을 때와 전화걸 때는 끄고, 응답할 때만 켠다.

ATOn - 온라인 상태 제어

온라인 데이터 상태로 어떻게 들어갈 지 결정한다.

ATO0 : 온라인 명령 상태일 때, 곧바로 온라인 데이터 상태로 들어간다. 온라인 명령상태가 아니라면 ERROR가 발생한다. 온라인 데이터 상태에서 이스케이프(+++)를 입력하면 온라인 명령 상태로 잠시 빠져나오는데, 이 상태에서 ATO0를 입력하면 곧바로 다시 온라인 데이터 상태로 돌아간다.

ATO1 : 온라인 명령 상태일 때, 약간의 시간 지연 후에 온라인 데이터 상태로 들어간다.

ATQn - 결과 코드 제어

결과 코드를 DTE에 보낼 것인지를 결정한다.

ATQ0 : 결과값을 DTE로 보낸다. 일반적으로는 생각하면, 화면에 출력되게 된다. 이 값이 기본값이다.

ATQ1 : 결과값은 DTE로 보내지 않는다. 일반적으로 생각하면, 화면에 결과값이 나타나지 않게 한다.

ATSn - S레지스터 값 읽고 쓰기

n은 생략할 수 있는데, 생략했을 때는 마지막에 접근한 레지스터를 대상으로 읽기 쓰기를 행한다. 예를 들어서 'ATS7 = 40'이라고 해서 S7 레지스터에 40을 넣어 주었다면, 이 후 그냥 'AT=20' 또는 'ATS=20'이라고 하면 S7레지스터에 40을 넣어 주는 것이 된다. 그리고 그냥 'ATS7'이라고만 하면 S7 레지스터에 어떤 값을 넣는 것은 아

니고, 단지 S7 레지스터를 마지막에 접근한 레지스터로 만든다. 이 후 'AT=40' 또는 'ATS=40' 하면 S7 레지스터에 값을 써 넣는 것이 된다.

ATSn : S레지스터 n을 마지막에 접근한 레지스터로 만든다.

ATSn=v : S레지스터 n에 v라는 값을 넣는다.

ATSn=? : S레지스터 n의 값을 출력한다.

ATVn – 결과값 형태

모뎀의 결과값은 우리가 흔히 보는 'OK', 'CONNECT xxxx', 'NO CARRIER'같은 영문 형태 말고 0, 1, 2, 3 같은 숫자 형태로도 얻을 수 있다. 보통 영문 형태를 긴 형태(long form), 숫자 형태를 짧은 형태(short form)라고 한다.

ATV0 : 결과값을 짧은 형태, 즉 숫자 형태로 돌려 준다.

ATV1 : 결과값을 긴 형태, 즉 영문 형태로 돌려 준다.

ATXn – 결과 부호 선택

얻을 수 있는 결과 부호의 범위를 선택한다. 기본 결과 부호만을 사용할 지, 확장 결과 부호를 사용할 지 결정한다.

ATX0 : OK, CONNECT, RING, NO CARRIER, ERROR, NO ANSWER 결과 부호만 얻을 수 있으며, 통화 중일 때는 NO CARRIER가 결과 부호값이다. 발신음(dial tone)이 없을 때도 NO DIALTONE 대신 NO CARRIER가 결과 부호이다.

ATX1 : OK, CONNECT, RING, NO CARRIER, ERROR, NO ANSWER, CONNECT xxxx가 결과 부호값이다. 통화 중일 때는 NO CARRIER가 결과 부호값이다. 발신음(dial tone)이 없을 때도 NO DIALTONE 대신 NO CARRIER가 결과 부호이다.

ATX2 : OK, CONNECT, RING, NO CARRIER, ERROR, NO ANSWER, CONNECT xxxx가 결과 부호값이다. 통화 중일 때는 NO CARRIER가 결과 부호값이다. 발신음(dial tone)이 없을 때도 NO DIALTONE 대신 NO CARRIER가 결과 부호이다.

ATX3 : OK, CONNECT, RING, NO CARRIER, ERROR, NO ANSWER, CONNECT xxxx, BUSY, NO DIALTONE이 결과 부호값이다.

ATX4 : 결과 부호값을 모두를 보고한다.

다음 표는 결과값 형태(ATVn)와 확장 결과 부호(ATXn) 중 일부를 나타낸 것이다.

짧은 형태 ATV0	긴 형태 ATV1	ATXn에서 n값				
		0	1	2	3	4
0	OK	◎	◎	◎	◎	◎
1	CONNECT	◎	◎	◎	◎	◎
2	RING	◎	◎	◎	◎	◎
3	NO CARRIER	◎	◎	◎	◎	◎
4	ERROR	◎	◎	◎	◎	◎
5	CONNECT 1200	1	◎	◎	◎	◎
6	NO DIALTONE	3	3	◎	◎	◎
7	BUSY	3	3	3	◎	◎
8	NO ANSWER	◎	◎	◎	◎	◎
9	CONNECT 0600	1	◎	◎	◎	◎
10	CONNECT 2400	1	◎	◎	◎	◎
11	CONNECT 4800	1	◎	◎	◎	◎
12	CONNECT 9600	1	◎	◎	◎	◎
13	CONNECT 7200	1	◎	◎	◎	◎
14	CONNECT 12000	1	◎	◎	◎	◎
15	CONNECT 14400	1	◎	◎	◎	◎
16	CONNECT 19200	1	◎	◎	◎	◎
17	CONNECT 38400	1	◎	◎	◎	◎
18	CONNECT 57600	1	◎	◎	◎	◎
19	CONNECT 115200	1	◎	◎	◎	◎
22	C O N N E C T 75TX/1200RX	1	◎	◎	◎	◎
23	C O N N E C T 1200TX/75RX	1	◎	◎	◎	◎
24	DELAYED	4	4	4	4	◎
32	BLACKLISTED	4	4	4	4	◎
33	FAX	◎	◎	◎	◎	◎
35	DATA	◎	◎	◎	◎	◎
40	CARRIER 300	◎	◎	◎	◎	◎
44	CARRIER 1200/75	◎	◎	◎	◎	◎
45	CARRIER 75/1200	◎	◎	◎	◎	◎
46	CARRIER 1200	◎	◎	◎	◎	◎
47	CARRIER 2400	◎	◎	◎	◎	◎
48	CARRIER 4800	◎	◎	◎	◎	◎
49	CARRIER 7200	◎	◎	◎	◎	◎
〈중략〉						

ATZn – 모뎀 초기화

모뎀을 소프트웨어적으로 초기화 한다. n을 생략하면 0으로 간주한다. 따라서 보통은 'ATZ'로 모뎀을 소프트웨어적으로 초기화한다.

ATZ0(=ATZ) : 모뎀을 초기화하고, 0번 프로파일에 저장된 값으로 복원한다.

ATZ1 : 모뎀을 초기화하고 1번 프로파일에 저장된 값으로 복원한다.