

**Komendy AT modemów  
analogowych, ISDN, GSM i SDI**

## Wstęp:

W książce poruszony będzie temat sterowania modemem przy użyciu jakiegokolwiek procesora i to zarówno od strony programowej jak i sprzętowej. Opisane będą dokładnie standardowe komendy AT oraz ich rozszerzenia. Wiedza ta może być przydatna konstruktorom urządzeń współpracujących z modemami jak i osobom konfigurującym modemy komputerowe.

Opisane będą programy i użytkowanie modemów z platformami:

- Win dla PC (komputery osobiste)
- DOS PC (sterowniki urządzeń)
- **Linux (łącza zapasowe w serwerach, komputery osobiste)**
- AmigaOS (serwery, komputery osobiste)

Modem (MODulator-DEModulator) - to urządzenie, którego zadaniem jest konwersja sygnału cyfrowego na analogowy oraz sygnału analogowego w cyfrowy. Modemy analogowe służą do przesyłania danych cyfrowych z wykorzystaniem analogowych łączy telefonicznych. Łącza takie przeznaczone były pierwotnie przeznaczone do transmisji audio (głosu). Aby więc umożliwić transmisję sygnału cyfrowego pomiędzy komputerami, poprzez łącza telefoniczne, konieczna jest zamiana sygnału cyfrowego w analogowy podczas nadawania. Proces taki odbywa się poprzez modulację sygnału i nazywany jest modulacją. Po stronie odbiorczej przeprowadzana jest demodulacja, zamiana sygnału analogowego na cyfrowy. Proces zamiany sygnału cyfrowego na analogowy.

Modemy, które komunikują się za pośrednictwem analogowych łączy telekomunikacyjnych muszą stosować takie same techniki komunikacyjne jak dla połączeń audio. Do najważniejszych z nich należą standardy CITT (ang. Consultative Committee for International Telegraph and Telephone). Drugi popularnym standardem jest MNP (ang. Microcom Networking Protocol).

W dzisiejszych czasach, w modemach cyfrowych (ISDN, DSL) nie zachodzi konieczność zamiany sygnałów z/do komputera na analogowe, ze względu na to, że cała transmisja przebiega w sposób cyfrowy. Można by sprzeczać się czy tor transmisyjny pomiędzy centralą a modemem jest cyfrowy czy analogowy ponieważ nie ma tam jak w typowej transmisji cyfrowej dwóch stanów lecz kilka lub nawet kilkadziesiąt. Jednak te kilkadziesiąt poziomów sygnału z pewnością nie można zaliczyć do transmisji analogowej, ponieważ dynamika tego sygnału analogowego była bardzo mała, a co za tym idzie mowa niezrozumiała..

## Rys historyczny

### History of the AT Command Set

In the early 1980s, a company called Hayes began manufacturing the Hayes Smartmodem 1200. Shortly after that, Hayes was forced to quickly release the Smartmodem 2400, as the market for 1200 baud/bps modems evaporated almost overnight. In the rush to get the Smartmodem 2400 completed and onto the market, there was no time for design alterations or embellishments between these two modem designs. This inadvertently created a new era in modem standardization that continues today.

The fact of the matter is that the claim that these two modems were "smart" was somewhat feeble, even compared to modems of that day. The Smartmodem 1200 and Smartmodem 2400 could dial telephone numbers by themselves, rather than the typical situation of expecting a human to dial the number on an attached telephone, and when the human heard the remote modem answer carrier, the human would press a "start" button on the modem and hang-up the extension telephone, allowing the two modems to establish a connection. The manual dialing process was fairly common in modems that had been manufactured for use with POTS/PSTN (Plain Old Telephone Service/Public Switched Telephone Network) lines up to the early 1980s. Automatic dialing modems were fairly rare, or the ability for a modem to dial telephone numbers was an extra-cost option. As an example, in 1983 the Radio Shack 1200baud/bps DC-1200 (a relabeled Ventel modem) cost \$699 and the auto-dial module cost an additional \$150, which could only do pulse dialing. Even the ability to disconnect the data connection frequently required the human to turn the modem off or place it in an on-hook state.

The Hayes Smartmodems also came with speaker, but then you really need the modem to have a speaker if you are going to let it dial numbers, so this probably really should not count as a feature. The Smartmodem 1200 and Smartmodem 2400 didn't know how to recognize a busy signal or the detect the presence of dial-tone, nor any of the dozens of other features now taken for granted in modems produced since the late 1980s.

Now, the Hayes Smartmodems did have support for several specialized applications, like the ability to transmit morse code, work with RTTY and work with amateur radio repeaters, but for people who just wanted to make a

data connection, these extras weren't interesting, and most of these special (or hobbyist) features disappeared in later years and hardly any modem maker copied these capabilities.

What did make the Smartmodem 1200 and Smartmodem 2400 unique at that time was that they were the first case of a vendor producing two different speed models of POTS/PSTN modems that accepted the same programming commands. As mentioned above, this was more of an accident than intentional, with Hayes doing a rushed (and minimal) rework of the Smartmodem 1200 design to support the 2400bps speed that the consumer modem market suddenly demanded. However, this did mean that any software that knew how to get a modem to dial a telephone number that worked on the Smartmodem 1200 would also work on the Smartmodem 2400.

At this point in time, there were relatively few commercial dial-up computer services, with CompuServe being the largest by far. Each time a vendor brought out a new modem, it was routine to wait some months for an updated driver to be written to allow a given brand of computer to be able to control the new type of modem and allow that computer and modem to connect to CompuServe. Of course, by the time the Smartmodem 2400 came on the market, there was already a working driver for the Smartmodem 1200 for many computer platforms, so people who bought the Smartmodem 2400 didn't have to wait for the driver to be written; they could use the one written for the Smartmodem 1200. Supporting the Smartmodem 2400 required little more than placing an addendum in the documentation stating that the Smartmodem 2400 would work using the Smartmodem 1200 driver.

Quite a few modem makers suddenly realized the advantage of making newer models of modems have the same command set as previous models, something that few vendors had ever bothered with, and even individual modem designers would routinely develop a completely different command set for each successive modem model that they created. The Smartmodem 1200 and 2400 largely ended that practice throughout the industry, and within six months, several vendors were offering "Hayes compatible" command sets as an option or the only choice on their modems.

Hayes promptly sued these companies for using the word "Hayes" in their product literature and packaging, so collectively, the other makers started calling their modems "AT Command Set compatible" and continued to mimic the Hayes command set. By 1986, essentially no consumer modems were made that didn't at least have an AT command set emulation mode.

Years later, the TIA/EIA introduced a formal standard with the title "Data Transmission Systems and Equipment - Serial Asynchronous Automatic Dialing and Control", otherwise known as TIA/EIA-602. TIA/EIA-602 is almost identical to the data-specific commands found in the Smartmodems 1200 and Smartmodem 2400. Of course, by the time the TIA/EIA-602 standard came out, vendors were selling modems with error-correction, compression and far higher speeds. None of these newer capabilities (or the commands needed to control them) are addressed by the TIA/EIA-602 standard, although other standards or drafts of standards exist for commands specific to FAX operations on modems that support FAX transmission or reception, as well as commands specific to voice operations.

For the newer modem-specific features, many manufacturers elected to copy the extensions to the original Hayes command set that appeared in the Hayes Ultra 14,400bps modems. These extensions are largely what Rockwell came up with, who was the primary provider of datapumps to Hayes at that time. Not all modem chipset makers followed the Ultra extended commands to the letter, resulting in some incompatibility that still exists in the more sophisticated commands. These incompatibilities normally surface when the chipset maker isn't Rockwell or a licensee of Rockwell's modem code. However, even licensees have altered what Rockwell distributes and in some cases Rockwell has introduced incompatibilities on their own.

### **Typy modemów**

Modemy możemy podzielić ze względu na sposób transmisji, typ portu do współpracy z komputerem, przeznaczenie.

### ***modemy analogowe***

W modemach tych do transmisji danych wykorzystuje się tor analogowy o paśmie przenoszenia 300Hz-3400Hz. W pierwszych standardach transmisji bit o wartości 1 reprezentowała jedna częstotliwość, o wartości 0 inna. Szybsze modemy działają podobnie do styku U w transmisji ISDN. W jednej jednostce czasu koduje się kilka

bitów danych. Współczesne modemy mogą przysyłać dane z prędkością dochodzącą do 57kb/s. Należy zaznaczyć, że transmisja jest asymetryczna (odbior 57kb/s, nadawanie 33,6kb/s).

### **modemy cyfrowe**

Do transmisji wykorzystana jest linia cyfrowa. W modemach **ISDN** wyróżnia się dwie pary (nadawczą i odbiorczą). Dane można przysyłać po jednym kanale z prędkością równą 64kb/s, po dwóch 128kb/s. Takie sygnały występują pomiędzy urządzeniem ISDN (telefon, modem) a terminalem (styk S/T). Połączenie pomiędzy centralą a terminalem wykonane jest przewodem jednoparowym (styk U).

W modemach **SDI** transmisja przebiega dokładnie tak samo jak w ISDN. Dane są transmitowane w dwóch kanałach z prędkością 128kb/s. Podczas rozmowy telefonicznej zwalniany jest jeden kanał na rozmowę. Prędkość transmisji spada wówczas do 64kb/s, po drugim kanale przesyłana jest cyfrowo rozmowa. Modemy SDI przeznaczone są głównie do połączeń internetowych)

W modemach **ADSL** dane cyfrowe przesyłane są w paśmie ponadakustycznym. Modemy te mają dość duże prędkości transmisji (np. Neostada 512kb/s) jednak są niesymetryczne, tzn. prędkość nadawania jest inna niż odbioru. Czasem na modemach ADSL buduje się łącza dzierżawione). Po tej samej linii co dane cyfrowe w paśmie akustycznym może być przesyłana rozmowa telefoniczna analogowa.

Modemy **SHDSL** są modemami symetrycznymi, tzn. prędkość nadawania i odbioru są sobie równe.

W modemach **GSM** wykorzystuje się transmisję cyfrową drogą radiową. Niestety transmisja przebiega z małą prędkością 9600b/s w pewnych warunkach dochodzi do 14,4kb/s.

### **modemy zewnętrzne**

Mogą być wyposażone w port RS232C lub popularny ostatnio port USB. Modemy ISDN są często wyposażone w szybki port RS232C (460kb/s a nawet blisko 1Mb/s). Niestety popularne PC nie są w stanie pracować z taką prędkością, nie ma natomiast problemu aby taką prędkość uzyskać na komputerach MAC czy Amiga.

### **modemy wewnętrzne**

Najczęściej spotykane są karty PCI. Dawniej popularne były karty na złącze ISA. Karty te mogą być atrakcyjne dla amatorów ze względu na niską cenę i prosty sposób podłączenia do dowolnego mikroprocesora (nie dotyczy WinModemów).

Modemy wewnętrzne są dwojakiego rodzaju:

Jedne symulują port COM i zawierają prawie takie same układy jak modemy zewnętrzne.

Inne (tzw WinModemy) są tylko interfejsem pomiędzy magistralą procesora a linią telefoniczną. W tych ostatnich transmisją znajduje się procesor komputera.

Zaletą modemów wewnętrznych jest ich niska cena, i mniejsza liczba kabli połączeniowych. Wad mają znacznie więcej:

- są bardziej kapryśne w uruchamianiu i użytkowaniu,
- najczęściej modemy analogowe (a znane mi ISDN wszystkie) są wykonywane jako tzw WinModemy. Jeśli nie posiadamy sterownika dla używanego przez nas systemu operacyjnego modemu takiego nie da się wykozystać. Ponadto procesor komputera musi zajmować się transmisją danych, co przy dodatkowej kompresji transmitowanych danych nie jest bez znaczenia dla wydajności komputera,
- są mniej odporne na uszkodzenia pochodzące z linii telefonicznej
- w wersji ISDN nigdy nie są wyposażone w porty POST (analogowe) i nie mogą spełniać dodatkowych funkcji poza transmisją danych.

### **Kable połączeniowe**

Modem należy podłączyć do komputera i linii telefonicznej. W większości modemów zewnętrznych wymagane jest jeszcze napięcie zasilające. Nie dotyczy to sporej części modemów USB, które zasilanie czerpią z magistrali.

## Kable komputerowe

### Modem wewnętrzny

Przy podłączaniu modemu wewnętrznego ominię nas konieczność połączenia modemu z komputerem. Wystarczy umieścić kartę modemu w odpowiednim slotcie komputera.

### Modem RS232C

Jeśli podłączamy modem zewnętrzny wyposażony w port RS232C musimy zaopatrzyć się w stosowny kabelek. Przeważnie jest on dostarczony wraz z modemem. Komputery najczęściej są wyposażone we wtyk DB9, natomiast modemy gniazdo DB25. Kupując modem z „drugiej ręki” może on nie być wyposażony w odpowiednie kabelki. Czasem producent dostarcza kabel w wtykiem DB9 od strony komputera, a jedyny wolny port szeregowy ma wtyk DB25. W tej sytuacji musimy zakupić odpowiedni kabelek lub wykonać go sami. W tabelkach poniżej pokazano sposób połączenia gniazda i wtyku we wszystkich możliwych kombinacjach gniazd i wtyków od strony komputera jak i modemu:

<b>Nazwa sygnału</b>	<b>Wtyk DB9 (do komputera)</b>	<b>Gniazdo DB25 (do modemu)</b>
Transmit Data	3	2
Receive Data	2	3
Request to Send	7	4
Clear to Send	8	5
Data Set Ready	6	6
System Ground	5	7
Carrier Detect	1	8
Data Terminal Ready	4	20
Ring Indicator	9	22

<b>Nazwa sygnału</b>	<b>Wtyk DB9 (do komputera)</b>	<b>Gniazdo DB9 (do modemu)</b>
Transmit Data	3	3
Receive Data	2	2
Request to Send	7	7
Clear to Send	8	8
Data Set Ready	6	6
System Ground	5	5
Carrier Detect	1	1
Data Terminal Ready	4	4
Ring Indicator	9	9

<b>Nazwa sygnału</b>	<b>Wtyk DB25 (do komputera)</b>	<b>Gniazdo DB25 (do modemu)</b>
Shield Ground	1	1
Transmit Data	2	2
Receive Data	3	3
Request to Send	4	4
Clear to Send	5	5
Data Set Ready	6	6
System Ground	7	7
Carrier Detect	8	8
Data Terminal Ready	20	20
Ring Indicator	22	22

<b>Nazwa sygnału</b>	<b>Wtyk DB25 (do komputera)</b>	<b>Gniazdo DB9 (do modemu)</b>
Transmit Data	2	3
Receive Data	3	2
Request to Send	4	7
Clear to Send	5	8

Data Set Ready	6	6
System Ground	7	5
Carrier Detect	8	1
Data Terminal Ready	20	4
Ring Indicator	22	9

Niektóre modemy są wyposażone w kabel uniwersalny ze złączami DB9 i DB25 od strony komputera. Nie oznacza to, że do jednego modemu można bezpośrednio podłączyć dwa komputery.

## Modem USB

Jeśli podłączamy modem USB potrzebny jest typowy kabel USB taki sam jak dla innych urządzeń (skaner, drukarka).

## Modem z portem równoległym

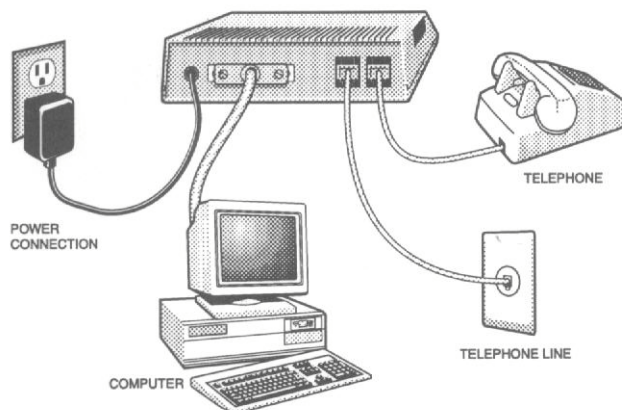
Zdają się także modemy z wejściem równoległym. Łączymy je najczęściej kabelkiem z wtyczką DB25 od strony komputera i gniazdem DB25 od strony modemu. Kabel jest wykonany przez połączenie ze sobą styków w gniazdach o tych samych numerach (tzn pin 1 gniazda do pinu 1 wtyku, pin 2 gniazda do pinu 2 wtyku, itd.). Styki 18...25 stanowiące masę mogą być ze sobą zwarte i połączone jednym przewodem.

## Kable telefoniczne

Aby modem mógł się połączyć z innym modemem musi być podłączony do linii telefonicznej.

## Podłączenie modemu analogowego

Problemy mogą pojawić się z podłączeniem linii telefonicznej. Większość modemów posiada dwa gniazda oznaczone jako „Tel” lub „Phone” oraz „Line”. Bardzo często modem jest tak zbudowany, że w stanie spoczynku modemu linia przechodzi z gniazda „Line” do „Phone”. W czasie pracy modemu (transmisji danych, nawiązywaniu połączenia) sygnał dochodzący do gniazda „Phone” jest odcinany. Dzięki temu podniesienie słuchawki nie powoduje zakłóceń w transmisji danych. Gdy posiadamy taki modem najlepiej aby był on

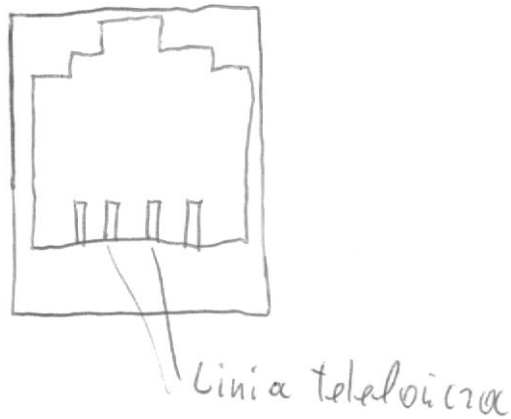


pierwszym urządzeniem telekomunikacyjnym

Rysunek „R1.jpg”

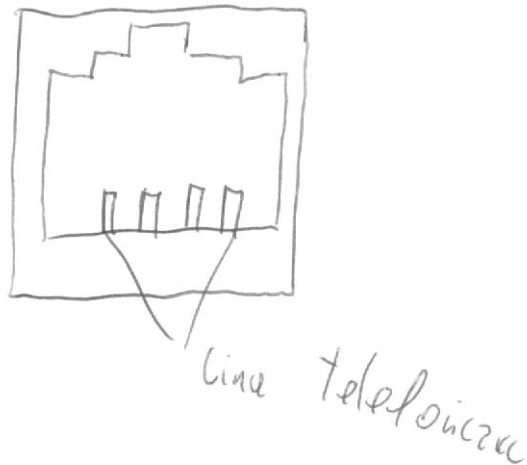
Bywają modemy w których oba gniazda są ze sobą połączone. W tej sytuacji podobnie zresztą jak i w tej w której modem jest wyposażony w jedno gniazdo, umiejscowienie modemu nie ma znaczenia, ponieważ nie potrafi on odłączyć dodatkowych urządzeń podłączonych do linii telefonicznej.

Podczas podłączania modemu można natknąć się na ciekawy problem. W większości państw linia telefoniczna jest doprowadzona do środkowych zacisków złącza RJ



rysunek „R2.jpg”

Producenci modemów budują je tak, że można je podłączyć kablem z wtykami RJ połączonym 1-1 (bez skrzyżowań żył). W Niemczech przyjętą się standard gdzie linia jest doprowadzona do skrajnych zacisków



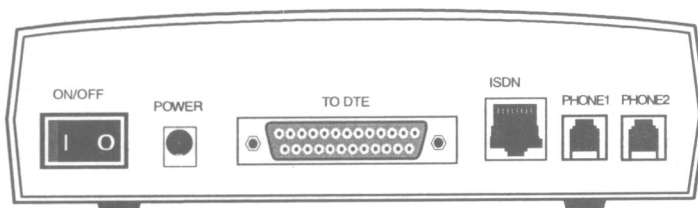
rysunek „R3.jpg”

i najczęściej modemy z tegoż kraju są zbudowane tak, żeby można było je połączyć kablem 1-1. Problem powstaje gdy chcemy podłączyć modem z Niemczech do linii w Polsce. Trzeba wykonać specjalny kabelek w którym para środkowa jest skrzyżowana z parą zewnętrzną. Podobny kłopot dotyczy także innych urządzeń (telefony, faksy) produkowanych w Niemczech.

Po podłączeniu wymagana jest instalacja sterowników w komputerze (W PC wymagana w Amigach i MAC'ach niekoniecznie). Procesu instalacji nie będę opisywał ponieważ jest on zależny od typu modemu i systemu operacyjnego, a jest on opisany w instrukcji obsługi modemu.

### Podłączenie modemu ISDN

Połączenie modemu ISDN jest tak proste jak i modemu analogowego. Linie telefoniczną ISDN łączymy typowym kabełkiem z wejściem ISDN w modemie. Na

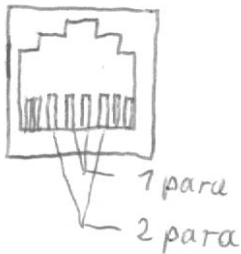


rysunku „r8.jpg”

jest to trzecie gniazdo od lewej.

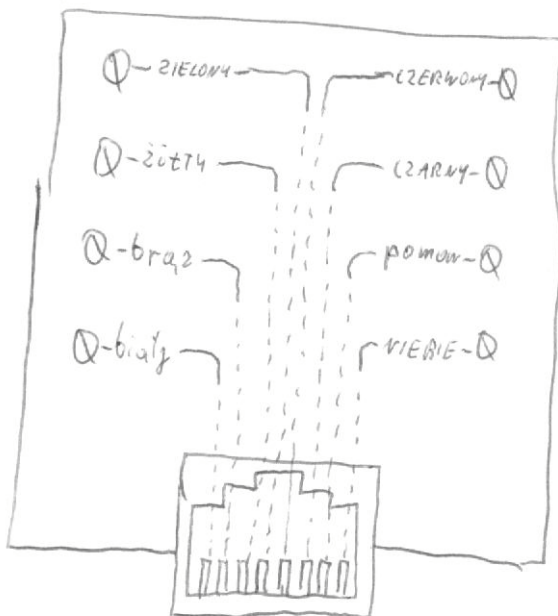
Zdarzają się modemy ISDN, które posiadają dwa wyjścia analogowe do podłączenia faksu czy telefonu analogowego. W tym przypadku urządzenia te podłączamy kablem telefonicznym analogowym do gniazd PHONE1 i PHONE2 (dwa pierwsze gniazda od prawej na rysunku „r8.jpg”).

Istotne jest podłączenie do terminala ISDN zwłaszcza gdy równoległe z modemem pracują inne urządzenia ISDN. Przy transmisji cyfrowej wykorzystywane są dwie pary nadawcza i odbiorcza jakkolwiek gniazda i wtyki mają po cztery pary (osiem zacisków). Jedna z par jest doprowadzona do środkowych zacisków gniazda, druga do kolejnych



rysunek „R4.jpg”

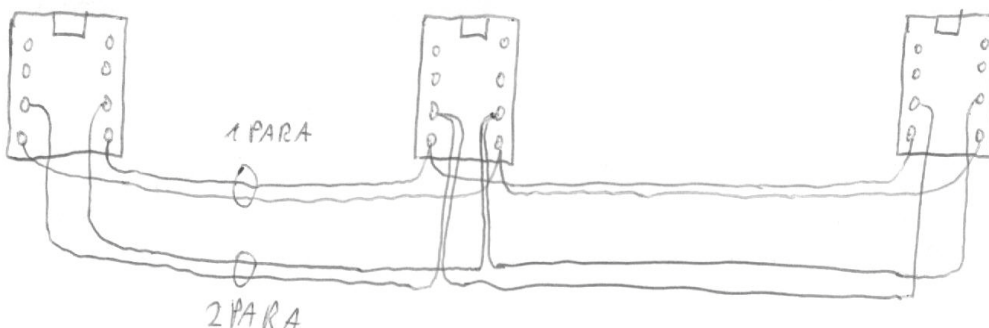
Zasadniczo polaryzacja żył w parze nie ma znaczenia, ważne aby nie zamienić pary nadawczej z odbiorczą (w przypadku zamiany nic się nie uszkodzi tyle, że mogą przestać działać urządzenia przyłączone do terminala). Uwaga ta jedna nie dotyczy sytuacji gdy do linii ISDN podłączone jest więcej urządzeń. Wtedy to polaryzacja żył ma znaczenie. Na



rysunku „R5.jpg”

opisano sposób połączenia zacisków w gnieździe z wkładem RJ. Jak łatwo zaobserwować pary wykorzystywane przez ISDN to para zielona/czerwona i żółta/czarna. Podczas prowadzenia instalacji należy więc uważać aby we wszystkich gniazdach kolor był połączony z kolorem oraz aby para przewodów tworzyła parę w kablu.

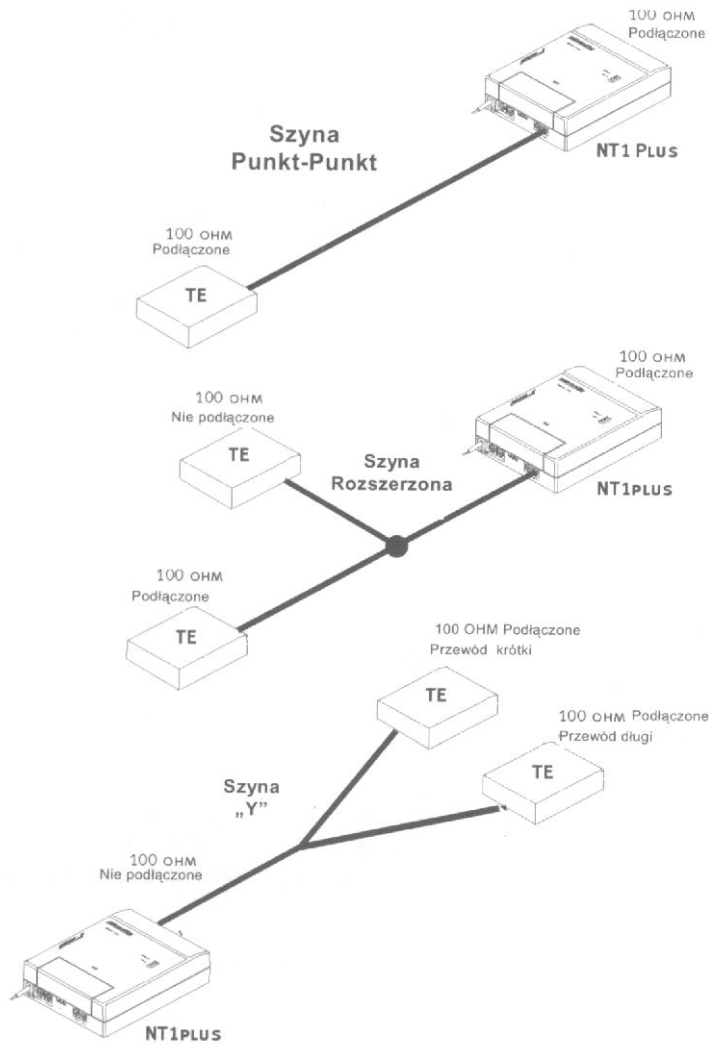
Rozparowanie przewodów, zwłaszcza na dłuższej linii, może spowodować zakłócanie się pary odbiorczej sygnałem nadawanym co zaowocuje zwiększeniem stopy błędów a co za tym idzie niższą jakością połączenia lub nawet jego brakiem. Prawidłowe połączenie gniazd przedstawia





rysunek „R6.jpg”.

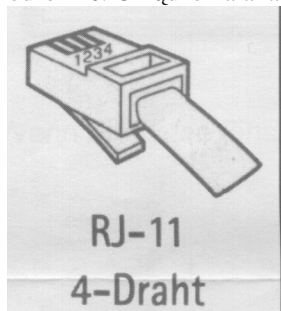
Pozostaje jeszcze problem terminowania linii gdy jest ona długa (dłuższa niż 10m). W takim przypadku na końcach linii włącza się rezystory o wartości 100ohm zarówno na parę nadawczą jak i odbiorczą. O tym w którym miejscu należy włączyć terminatory można dowiedzieć się z rysunku



rysunek „R7.jpg”.

## Różnice w kablach analogowych i ISDN

Kable i gniazda analogowe i cyfrowe w urządzeniach telekomunikacyjnych mają różne wtyki, aby było je łatwo odróżnić. Urządzenia analogowe są wyposażone w złącza typu RJ-11



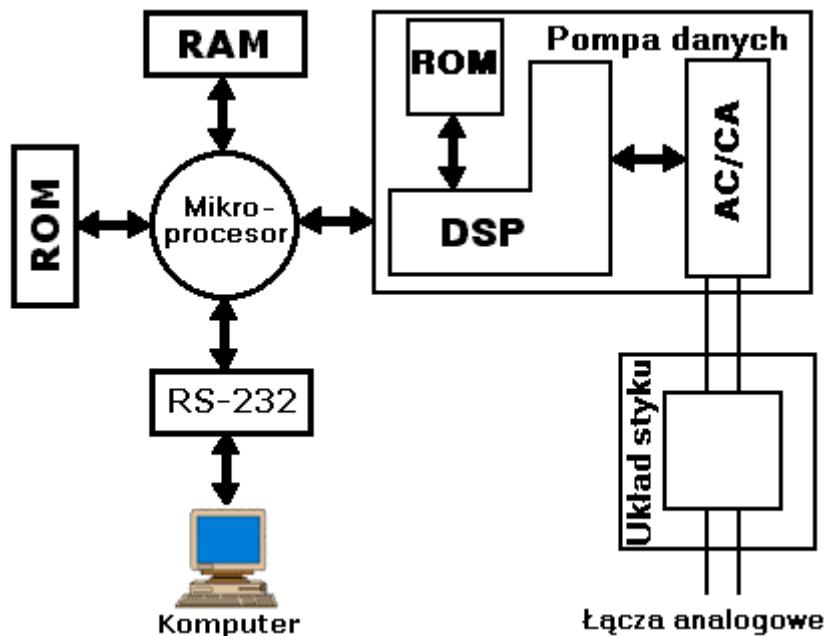
rysunek „r9.jpg”

i do poprawnej pracy wystarczy jedna para przewodów (dwa „druty” oznaczone na rysunku cyframi 2 i 3).

Urządzenia ISDN wyposażone są w złącza typu RJ-45 (jak w sieciach Ethernet) z tym, że do pracy wystarcza dwie pary przewodów.

## Budowa modemów

### Modemy analogowe



Rysunek „budowa.gif”

Budowa modemu:

Poniższy rysunek przedstawia schemat poglądowy pozwalający poznać relacje pomiędzy poszczególnymi składowymi modemem.

Opis schematu:

\* Pompa danych (ang. Data Pump) - której zadaniem jest modulacja/demodulacja sygnałów. W jej środku znajduje się procesor sygnałowy DSP, który posiada zawarte w pamięci ROM określone procedury modulacji (np. zgodnie ze standardem CCITT), również przetworniki AC/CA, dzięki którym możliwa jest konwersja cyfra/analog i analog/cyfra. Ponadto zadaniem układu jest eliminacja różnego rodzaju zakłóceń.

\* Mikroprocesor - steruje całą pracą układu. Posługuje się on wewnętrznym językiem AT, za pomocą rozkazów którego możliwa jest komunikacja pomiędzy komputerem i modemem, wybieranie numeru adresata, kompresja danych i kodowanie zabezpieczające. Program sterujący mikroprocesorem znajduje się w postaci ROM. Procesor dysponuje również pamięcią podręczną RAM.

\* RS-232 - czyli układ transmisji szeregowej, zajmuje się transmisją danych pomiędzy komputerem a modemem.

\* Łącze - odpowiedzialne jest za pulsacyjne wybieranie numeru oraz izolację galwaniczną linii telefonicznej od modemu.

### Modemy cyfrowe

## Zasada modulacji sygnału w modemach

### Transmisja analogowa

Modem podłączony jest zazwyczaj do komputera złączem RS-232 i może być wykonany jako karta rozszerzająca (modem wewnętrzny), jak i osobne urządzenie montowane na zewnątrz komputera.

\* Łącza modemowe.

Pasma przenoszenia dla linii telekomunikacyjnych wynosi około 3000 Hz. Jest ono wystarczające do transmisji głosu. Podczas transmisji danych musimy wziąć pod uwagę fakt, że ze względu na różnice w jakości połączeń telefonicznych nie jest możliwe, aby całe pasmo było przeznaczone do transmisji danych. Dlatego też stosowane są pasma ochronne, których rola jest prosta: oddzielić częstotliwość nadawczą od odbiorczej (dla trybu duplex). Zadaniem pasma ochronnego jest zabezpieczenie sygnału przed zakłuceniami.

#### Łącza telefoniczne

Istnieją dwa rodzaje łączy pozwalających na komunikację modemową:

\* Łącze komutowane - połączenie odbywa się poprzez dzielenie łącza. Połączenie jest zestawiane pomiędzy dwoma adresatami. Zakończenie rozmowy powoduje likwidację łącza.

\* Łącze dzierżawione - różni się tym od łącza komutowanego, że nie występuje nawiązywanie i rozłączanie połączenia. Połączenie zestawiane jest na stałe.

#### Protokoły i standardy transmisji telefonicznej

Protokoły i standardy są zbiorem reguł i uzgodnień, według których mogą komunikować się modemy. Standaryzacją w dziedzinie modemów zajmuje się Międzynarodowa Unia Telekomunikacyjna (ang. International Telecommunications Union, ITU), która reprezentuje rządy, instytucje naukowe i organizacje badawcze z całego świata. Zagadnienia dotyczące modemów, ITU publikuje w serii zaleceń oznaczonych symbolem V.xx (np.: V.22). Jeśli producent informuje, iż jego modem pracuje zgodnie z protokołem V.22, oznacza to, że musi on spełniać wszystkie wymagania podane przez ITU w zaleceniu V.22, ponadto może współpracować tylko z takim modemem, który również realizuje to zalecenie. Protokoły ściśle definiują sposoby modulacji sygnałów przesyłanych pomiędzy komputerami, szybkości transmisji, sekwencje sygnałów w kanale telefonicznym podczas inicjowania transmisji itd. Dla każdego sposobu modulacji określona jest częstotliwość nośnej. Nośna to sygnał sinusoidalny o określonej częstotliwości. Częstotliwość ta musi dać się przesłać bez zniekształceń przez linię telefoniczną, czyli nie może być większa od 3,4 kHz. Dla każdego sposobu modulacji norma określa szybkość bitowego strumienia wejściowego. Dane zawarte w tym strumieniu bitowym zmieniają jeden lub kilka parametrów nośnej. Każdy sposób modulacji związany jest ściśle ze sposobem kodowania. Przed modulacją wejściowy strumień danych podlega pewnym procesom matematycznym - kodowaniu. Transmitowany ciąg bitów podlega zamianie na inny, dogodniejszy dla transmisji. Po stronie odbiorczej następuje demodulacja i dekodowanie zgodnie z procedurami dokładnie odwrotnymi do kodowania. Możemy wyróżnić następujące rodzaje protokołów dotyczących transmisji danych za pomocą faksów i modemów:

Zalecenie	Szybkość transmisji	Zastosowanie
V.21	300 b/s	modem
V.22	1200/6000 b/s	modem
V.22 bis	2400/1200 b/s	modem
V.23	1200/600 b/s	modem
V.27	4800/2400 b/s	fax
V.29	9600 b/s	fax
V.17	14400 b/s	fax
V.32	9600 b/s	modem
V.32 bis	14400 b/s	modem
V.34	28800 b/s	modem
V.34 bis	33600 b/s	modem
V.90	56 kb/s	modem
V.92	56 kb/s	modem

#### Protokół V.21

Obecnie rzadko stosowany, służy do transmisji asynchronicznej. Modem, który nawiązał połączenie generuje nośną  $f_n = 1080$  Hz - jest to sygnał pauzy. W takt napływającego strumienia bitowego zmienia się częstotliwość nośnej z szybkością 300 bps. Każdemu stanowi niskiemu - zeru odpowiada częstotliwość 1180 Hz, każdemu stanowi wysokiemu - jedynce odpowiada częstotliwość 980 Hz. Modem odbierający, do którego zadzwoniono, nadaje dwupleksowo modulując częstotliwość w granicach 1850 Hz dla zera i 1650 Hz dla jedynki. Sygnał pauzy to częstotliwość 1750 Hz. Owa modulacja częstotliwościowa nazywana jest w skrócie FSK (Frequency Shift Keying). Po stronie odbiorczej modem próbuje sygnał odebrany, eliminuje własny sygnał nadawany, przeprowadza detekcję częstotliwości i stara się zlokalizować w czasie chwilę zmiany częstotliwości nośnej odebranej z linii telefonicznej. Te chwile zmiany częstotliwości występują tylko wtedy, gdy w odbieranym ciągu bitowym występuje zmiana stanu niskiego na wysoki lub wysokiego na niski. Odbiornik - demodulator synchronizuje swój zegar do zlokalizowanych chwil czasu. Zegarem tym będzie taktowany odebrany asynchroniczny ciąg bitowy.

#### Protokół V.22

Modulacja stosowana w tym protokole nazywana jest w skrócie PSK (Phase Shift Keying). Polega ona na skokowej zmianie fazy nośnej w stosunku do fazy bieżącej. Protokół V.22 przewiduje szybkość modulacji 600 bodów (szybkość modulacji  $V_m$  podajemy w bodach, określa ona liczbę elementów przesyłanych w ciągu sekundy, przy szybkości transmisji np. 2400 b/s, gdy element sygnału utworzony jest przez cztery bity szybkość modulacji wyniesie 600 bodów). Wybrany parametr (faza) nośnej zmienia się 600 razy w ciągu sekundy. Nie oznacza to wcale, że szybkość przesyłanego strumienia bitowego będzie równa właśnie 600 bps. Przed modulacją warstwa przygotowania danych dokonuje skramblowania, to znaczy miesza wejściowy strumień danych z ciągiem zwiększającym częstotliwość zmian bitów na przeciwny i nie dopuszcza długich ciągów zer czy jedynek. Istnieją dwie możliwości modulacji. Pierwsza możliwość (wolniejsza) polega na tym, że warstwa przygotowania danych dostarcza dane bitowe z szybkością 600 bodów. Każdy bit równy zero zmienia fazę nośnej o  $90^\circ$ , każdy bit równy jeden zmienia fazę nośnej o  $270^\circ$ . Odbiornik, podobnie jak w protokole V.21, odtwarza zegar i dane, by przekazać je warstwie przygotowania danych, która wykonuje proces odwrotny do skramblowania, czyli deskrambling. Transmisja może być synchroniczna lub asynchroniczna. Druga możliwość polega na tym, że w nadajniku warstwa przygotowania danych dostarcza strumień bitowy dwa razy szybciej, z przepływnością 1200 bodów. Przychodzące bity po skramblowaniu w specjalnym rejestrze grupowane są w pary. Każdej parze przyporządkowano określoną zmianę fazy. Aby jednoznacznie określić parę bitów należy zdefiniować cztery stany: 00, 01, 10, 11. Każdemu stanowi przyporządkowano jedną zmianę fazy o wartość  $90^\circ$ ,  $0^\circ$ ,  $180^\circ$ ,  $270^\circ$ . Odbiornik wykrywa zmiany fazy, odtwarza zegar dwa razy szybszy i w jednej chwili czasu dekoduje nie jeden bit, ale dwa, stanowiące zakodowaną parę. W ten sposób do warstwy przygotowania danych wysyłany jest z demodulatora strumień dwa razy szybszy od szybkości modulacji. Mechanizm ten jest typowy dla wszystkich nowszych protokołów. W ten sposób możliwe stało się przesyłanie wąskopasmowym kanałem telefonicznym strumieni o wyższych przepływnościach. Transmisja między modemami jest dwupleksowa. Modem nawiązujący połączenie wykorzystuje nośną 1200 Hz, modem odbierający 2400 Hz.

#### Protokół V.22 bis

Jest to uzupełnienie protokołu V.22 o szybkość 2400 bps. Stosuje się te same nośne 1200 Hz i 2400 Hz, tę samą szybkość modulacji 600 bodów. Dane wyjściowe po skramblowaniu formowane są w czwórki bitów. Każdej czwórce, a może ich być 16 różnych, przyporządkowano jeden stan zmiany parametrów nośnej. Zmiana parametrów nośnej polega na równoczesnej zmianie fazy i amplitudy. Pierwsze dwa bity kodują zmianę fazy tak, jak w protokole V.22, tworząc cztery stany fazowe sygnału nośnej. Następne dwa bity kodują amplitudę. Możliwe są też cztery stany fazowe sygnału nośnej. W sumie 16 stanów. Ten typ modulacji nosi nazwę QAM (Quadrature Amplitude Modulation). Praktycznie sygnał nośny powstaje w wyniku sumowania dwóch przebiegów sinusoidalnych przesuniętych w fazie o  $90^\circ$ . Kluczowana jest równocześnie amplituda obu składowych.

#### Protokół V.23

Umożliwia przesyłanie danych dwupleksowo z szybkością 1200 bodów dla modemu odpowiadającego i 75 bodów dla wywołującego. Dla przesyłu półdupleksowego szybkość wynosi 1200 bps.

#### Protokół V.27

Stosowany do transmisji faksów. Prędkość transmisji  $V_{\text{tran}}=4800/2400$  b/s. Wykorzystywana dwupleksowa transmisja synchroniczna.

#### Protokół V.29

Kolejny z protokołów dotyczących transmisji danych za pomocą faksów. Oferuje szybkość transmisji 9600 b/s, podobnie jak V.27 korzysta z dwupleksowej transmisji synchronicznej.

#### Protokół V.17

Protokół stosowany do transmisji faksów. Prędkość transmisji wynosi: 14400 b/s, 12000 b/s, 9600b/s, 7200 b/s. Wykorzystywana jest modulacja QAM z synchroniczną zmianą parametrów nośnej z szybkością 2400 bodów,  $f_n=1800$  Hz. Przed modulacją wejściowy strumień bitów dzielony jest na grupy:

po 6b dla  $V_{\text{tran}}= 14400$  b/s

po 5b dla  $V_{\text{tran}}= 12000$  b/s

po 4b dla  $V_{\text{tran}}= 9600$  b/s

po 3b dla  $V_{\text{tran}}= 7200$  b/s

Dwa najmłodsze bity podlegają operacji skramblowania. Zabieg ten polega na zamianie pary bitów na inną parę wyjściową zależną od pary bitów wejściowych i poprzedniej pary bitów wyjściowych. Proces ten jest stosowany w celu uniknięcia długich czasów bez zmiany parametrów nośnej. Na podstawie pary bitów wyjściowych generowany jest dodatkowy bit umieszczany na najmłodszej pozycji grupy bitów (w ten sposób dodaje się 1b) i uzyskujemy 3b. Taka grupa bitów (3b) zmienia parametry nośnej, zarówno fazę jak i amplitudę tworząc:

128 różnych stanów dla  $V_{\text{tran}}=14400$  b/s

64 różne stany dla  $V_{\text{tran}}= 12000$  b/s

32 różne stany dla  $V_{\text{tran}}=9600$  b/s

16 różnych stanów dla  $V_{\text{tran}}=7200$  b/s

#### Protokół V.32

Oba modemy stosują tę samą nośną 1800 Hz. Sygnał modemu odległego uzyskuje się metodami numerycznymi poprzez eliminację sygnału własnego. Szybkość modulacji jest stała i wynosi 2400 bodów. Koduje się dwójki lub czwórki bitów uzyskując szybkości bitowe 4800 bodów lub 9600 bodów. Stosuje się modulację QAM w połączeniu z metodą kodowania TCM (Trellis Coded Modulation). Metoda TCM polega na zastąpieniu każdej kodowanej czwórki bitów kombinacją pięciu bitów, nie wykorzystując wszystkich 32 możliwości. Sposób kodowania dobrano w ten sposób, by zmniejszyć różnicę parametrów sygnału nośnej dla sąsiednich kombinacji. Wprowadza się w ten sposób korekcję polegającą na eliminowaniu stanów niemożliwych z założenia. Norma przewiduje, że w wyniku dużej stopy błędów system automatycznie zmieni szybkość bitową.

#### Protokół V.32 bis

Rozszerzenie normy V.32 o szybkości bitowe 7200 b/s, 1200 b/s i 14400 b/s. Wspólna nośna 1800 Hz, szybkość modulacji 2400 bodów, kodowane dwójki, trójki, czwórki i piątki bitów. Dla szybkości od 4800 b/s począwszy stosuje się kodowanie kratowe TCM. Przewiduje się zmianę szybkości bitowej w zależności od stanu linii. Dopuszcza się powtórny negocjacje szybkości.

#### Protokół V.34

Protokół modulacji umożliwiający szybkości bitowe 19200 b/s i 28800 b/s. Protokół opracowany przez firmę Rockwell. Umożliwia on stosowanie czteroprzewodowej linii dzierżawionej lub dwuprzewodowej linii komutowanej (normalna linia telefoniczna). Protokół umożliwia skuteczną kontrolę jakości transmisji, a na podstawie której zmieniana jest nie tylko szybkość modulacji, ale także częstotliwość nośnej.

#### Protokół V.34 bis

Protokół ten to rozszerzenie metod stosowanych w V.34 do szybkości bitowych 31200 b/s i 33600 b/s.

### Protokół V.90

Jest to obecnie najnowszy protokół modulacji, umożliwiający nawiązanie połączenia między modemami przy prędkości do 56 kbs. Protokół ten powstał w lutym 1998 roku i był kompromisem kończącym spór między firmami Rockwell/Lucent i 3COM/USR. Firmy te niezależnie opracowały swoje protokoły modulacji K56flex i X2. Istnienie dwóch różnych protokołów powodowało niemożliwość połączenia się przy prędkości 56 kbs przy stosowaniu modemów konkurujących ze sobą producentów. Obecnie do wszystkich modeli modemów pracujących z protokołem K56flex lub X3 dostępne są, na stronach producentów, nowe wersje oprogramowania, umożliwiające pracę z protokołem V.90. Aby możliwe było nawiązanie połączenia przy protokole V.90 między dwoma modemami, przez komutowaną sieć telefoniczną, wymagana jest linia dobrej jakości oraz odległość między modemami mniejsza od 3 km. Gdy odległość przekracza 3 km, prawdopodobieństwo nawiązania połączenia między modemami w protokole V.90 staje się niewielkie. Protokół V.90 wykorzystuje transmisję danych cyfrowych, używa modulacji impulsowo-kodowej (PCM). Jeżeli niemożliwe jest połączenie się w protokole V.90, modem automatycznie wybiera modulację V.34. V.90 umożliwia transmisję danych z szybkością 56 kb/s tylko w jednym kierunku- od serwera internetowego do użytkownika. Natomiast użytkownik może wysyłać dane z szybkością 33.6 kb/s.

### Protokół V.92

Główne zalety protokołu V.92 związane są z jego funkcjonalnością. Modemy wykorzystujące nowy standard mają znacznie szybciej nawiązywać połączenia dzięki bardziej efektywnemu systemowi uzgadniania warunków transmisji. Zapewnia to opracowana specjalnie na potrzeby standardu V.92 nowa technologia QuickConnect. Kolejną i chyba najważniejszą konsekwencją wdrożenia V.92 będzie możliwość przyjmowania głosowych połączeń telefonicznych podczas surfowania. Jest to funkcja przydatna szczególnie użytkownikom indywidualnym, łączącym się z Internetem z domu. Większość z nich ma tylko jedną linię telefoniczną i zajmowanie jej przez modem uniemożliwia równoczesne tradycyjne połączenia. Użytkownik modemu zgodnego z V.92 otrzyma powiadomienie o przychodzącym połączeniu i będzie mógł zawiesić przesyłanie danych na czas rozmowy (z możliwością powrotu do surfowania po sieci w dowolnej chwili). Pauza zostanie jednak prawdopodobnie ograniczona do około 15 minut. Zmiany, które wnosi V.92, dotyczą również wydajności modemów. Szybkość przesyłania (wysyłania) danych od użytkownika zwiększy się z 33,6 do 48 Kb/s, a dzięki zastosowaniu nowego systemu kompresji (V44) zwiększy się także ogólna prędkość przesyłania danych. Do korzystania z dobrodziejstw nowej technologii modemowej niezbędna będzie jednak modernizacja urządzeń dostępowych w centralach telefonicznych. Polscy operatorzy i dostawcy usług internetowych zapowiadają poczynienie odpowiednich kroków, jednak od zatwierdzenia V.92 minie jeszcze trochę czasu, zanim użytkownicy będą mogli odnieść wszystkie korzyści związane z wprowadzeniem nowego standardu. Mimo że specyfikacja V.92 funkcjonuje wciąż półoficjalnie, wielu producentów modemów zapewnia o wstępnej zgodności oferowanych urządzeń z tym standardem.

Należy również wspomnieć o innych standardach. Przed kilkunastoma laty, gdy koncern AT&T (American Telephone and Telegraph) był dominującą firmą telefoniczną, wszystkie modemy musiały spełniać standard BELL'a. Wraz ze zmniejszeniem dominacji AT&T, standardy BELL'a stały się również mniej dominujące. Jednakże nadal w wykazach specyfikacyjnych modemów można zobaczyć te standardy, gdyż definiują one sposób komunikacji z mniejszymi szybkościami szczególnie w USA. Inna grupa zaleceń definiuje parametry i procedury modemu nie związane z transmisją danych:

### Protokół V.24

Definiuje linie interfejsu szeregowego. Interfejs V.24 wyposażony jest w linie pozwalające zorganizować pętlę testującą: lokalną (Local Analog Loopback, LAL, linia 141) oraz zdalną (Remote Digital Loopback, RDL, linia 140). Ponadto V.24 posiada linię TI (142), zwaną wskaźnikiem testu, za pomocą którego modem sygnalizuje DTE (komputerowi) fakt wykonania testu.

### Protokół V.28

Określa charakterystyki prądowo-napięciowe interfejsu.

### Protokół V.42

Określa korekcję błędów. Norma V.42 przewiduje dwa protokoły: LAPM (Link Access Protocol for Modems) i MNP-4 jako dodatkowy. LAPM dzieli dane na bloki o wielkości zależnej od szybkości przesyłania danych i jakości linii oraz szybkości portu szeregowego. Każdy blok uzupełniany jest o tzw. sumę kontrolną CRC (Cyclic Redundancy Code - kod cyklicznej kontroli nadmiarowej), to jest dodatkowe bity wyliczane na podstawie bitów tworzących dany blok. W modemie odległym odbierane są bity danych, a na ich podstawie obliczany według tej samej funkcji ten sam CRC i porównywany z odbieranym. Wynik porównania jest transmitowany do nadajnika, a ten, w wypadku wystąpienia błędu, retransmituje cały blok. Warstwa przygotowania danych usuwa, z napływającego portem szeregowym strumienia danych, bity startu i stopu, ale dodaje bity CRC, co ostatecznie spowalnia transmisję danych około 10%. Im krótszy jest CRC w stosunku do odległości bloku tym transmisja jest szybsza, ale skuteczność korekcji mniejsza. MNP-4 działa na podobnej zasadzie: uzupełnianie bloków o CRC, potwierdzanie i retransmitowanie.

#### Protokół V.42 bis

W normie tej zawarto opis procedur kompresji danych. Przewiduje się stosowanie dwóch algorytmów BLTZ (British Telecom Lempe Ziv) i MNP-5. Metoda BLTZ polega na wielokrotnym stosowaniu procedury RLE. Metoda RLE przewiduje zastępowanie ciągów znaków innym ciągiem zgodnym ze słownikiem, który jest ciągle tworzony. W przypadku braku efektów kompresji procedura jest przerywana. W przypadku większości danych efektywność kompresji wynosi 4:1. Metoda MNP-5 to też procedura RLE, inny jest nieco sposób kodowania. Znak koduje się różną ilością bitów, zależną od częstości występowania znaku. Efektywność kompresji średnio wynosi 2:1.

#### Protokół V.44

Niedawno opracowany i zatwierdzony został nowy standard kompresji - V.44. Charakteryzuje się on znacznie lepszym współczynnikiem upakowania informacji niż V.32 bis. Niestety, V.44 związany jest z protokołem V.92, na którego wprowadzenie będziemy musieli jeszcze trochę poczekać.

#### Protokół V.54

Określa system testów modemu. Modemy mogą obsługiwać kilka różnych testów, zwanych pętlami.

Szumy i zniekształcenia wprowadzane przez linię telefoniczną są powodem poważnych problemów podczas transmisji danych. Nowoczesne modemy wyposażane są więc w sprzętową korekcję błędów, zgodną z protokołem V.42 (ITU) lub protokołami MNP (Microcom Networking Protocol) firmy Microcom. Podczas transmisji z korekcją błędów dane przesyłane są blokami o długościach od 64 do 256 znaków.

#### Protokół MNP-1

Dotyczy korekcji błędów podczas transmisji asynchronicznej znakowej w trybie półduplexowym. Efektywna szybkość transmisji wynosi tu 70%. (jeśli modemy pracują z szybkością 2400 b/s, to efektywna szybkość transmisji, tzn. szybkość między komputerami wynosi 1690 b/s).

#### Protokół MNP-2

Jest to również protokół transmisji asynchronicznej znakowej. Stosuje on jednak duplexowy tryb wymiany danych dzięki czemu jego efektywność wynosi około 84%. Modem o szybkości 2400 b/s pracujący z protokołem MNP-2 osiąga efektywną szybkość równą 2000 b/s.

#### Klasa 3 protokołu MNP

Jest protokołem transmisji synchronicznej bitowej. Z uwagi na to, że protokół synchroniczny umożliwia eliminację (niezbędnych w transmisji asynchronicznej) bitów "startu" i "stopu", protokół klasy 3 MNP jest bardziej efektywny. Oczywiście wymiana danych pomiędzy komputerem i modemem odbywa się w dalszym ciągu asynchronicznie, ale modemy przesyłają dane pomiędzy sobą w sposób synchroniczny. Protokół klasy 3 stosuje duplexowy sposób wumiany danych. Jest to pierwsza z klas MNP, która zapewnia zarówno korekcję błędów jak i skrócenie czasu transmisji. Efektywność wynosi tu około 108%. Używając tej klasy w modemie o szybkości 2400 b/s, osiąga się efektywne przesyłanie danych z szybkością 2600 b/s.

#### Klasa 4 protokołu MNP

Posiada cechy klasy 3 uzupełnione o dalsze udoskonalenia. Modem pracujący w klasie 4 MNP nieustannie śledzi liczbę błędów spowodowaną szumami linii telefonicznej. Jeśli linia telefoniczna umożliwia bezbłędną transmisję, modem automatycznie zwiększa rozmiar kolejnych bloków danych, tzn. jeśli przesyłane są bezbłędnie bloki np. 64-bajtowe, modem zwiększa ich długość do 128 bajtów, itd. Gdy jakość linii jest niska i pojawiają się błędy, rozmiar bloku jest zmniejszany, dzięki czemu skraca się sumaryczny czas retransmisji (ponownej transmisji bloków w których wykryto błędy). Mniejszy rozmiar bloku zwiększa prawdopodobieństwo bezbłędnej transmisji za pierwszym razem. Efektywność protokołu wynosi około 120%. Dzięki temu modem, pracując z MNP klasy 4 uzyskuje efektywną przepływność binarną 2900 b/s.

Modem pracujący z protokołem V.42 może transmitować dane z korekcją do modemu pracującego z protokołem MNP klasy 3 lub 4.

Współczesne modemy umożliwiają kompresję danych, dzięki której skrócony zostaje czas transmisji (skompresowane pliki mają mniejsze rozmiary). Oprócz protokołu V.42 bis kompresję danych opisuje także protokół MNP 5. Kompresja zgodna z V.42 bis może być zastosowana tylko wtedy, gdy wymiana danych kontrolowana jest przez protokół korekcji błędów V.42 (podobnie kompresja z protokołem MNP5 stosowana jest dla wymiany danych z korekcją MNP4). Skuteczność kompresji zależy od typu transmitowanego zbioru. Zbiory COM i EXE poddają się kompresji słabo. Najlepsze efekty uzyskuje się dla zbiorów tekstowych. Dzięki kompresji danych, protokół MNP5 zwiększa efektywną szybkość transmisji dwukrotnie, natomiast protokół V.42 bis- czterokrotnie.

Przykłady maksymalnych szybkości przepływu danych pomiędzy komputerami, dla różnych standardów modulacji, z zastosowaniem kompresji MNP5 i V.42 bis:

Maksymalna efektywna szybkość tr. danych komputer-komputer				
Standard modulacji	Szybkość transmisji modem-modem		MNP5	V.42 bis
V.22	1200 bit/s	2400 bit/s	4800 bit/s	
V.22 bis	2400 bit/s	4800 bit/s	9600 bit/s	
V.32	9600 bit/s	19200 bit/s	38400 bit/s	
V.32 bis	14400 bit/s	28800 bit/s	57600 bit/s	
V.34	28800 bit/s	57600 bit/s	115200 bit/s	

Najnowszy protokół MNP 10 pozwala zrealizować połączenia dotychczas niemożliwe. Został on opracowany w zasadzie dla telefonii komórkowej, chociaż może być przydatny również dla łącz kablowych o niskiej jakości. Protokół ten charakteryzują następujące cechy:

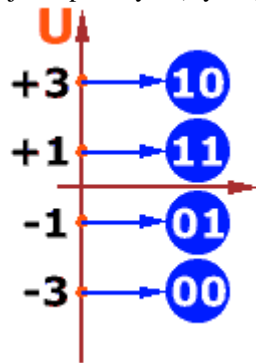
- negocjowanie i zwiększanie szybkości transmisji (modem rozpoczyna transmisję od niskiej szybkości, a następnie zwiększa ją do wartości na którą pozwalają warunki panujące w linii),
- wymuszanie połączenia (modem łączy się nawet wtedy, gdy w linii występują zakłócenia),
- dynamiczne zmiany prędkości (w czasie trwania połączenia, modem stale dopasowuje szybkość transmisji do warunków panujących w linii),
- szybka zmiana długości ramki (rozmiar ramki w niesprzyjających warunkach może zmieniać się od 256 bajtów do 8 bajtów),
- dopasowanie poziomu nadawanego sygnału (modem ustala odpowiedni poziom nadawanego sygnału dla warunków panujących w linii).

#### **Transmisja cyfrowa**

W modemach ISDN wyróżnia się dwie pary (nadawczą i odbiorczą). Dane można przysyłać po jednym kanale z prędkością równą 64kb/s, po dwóch 128kb/s. Takie sygnały występują pomiędzy urządzeniem ISDN (telefon,

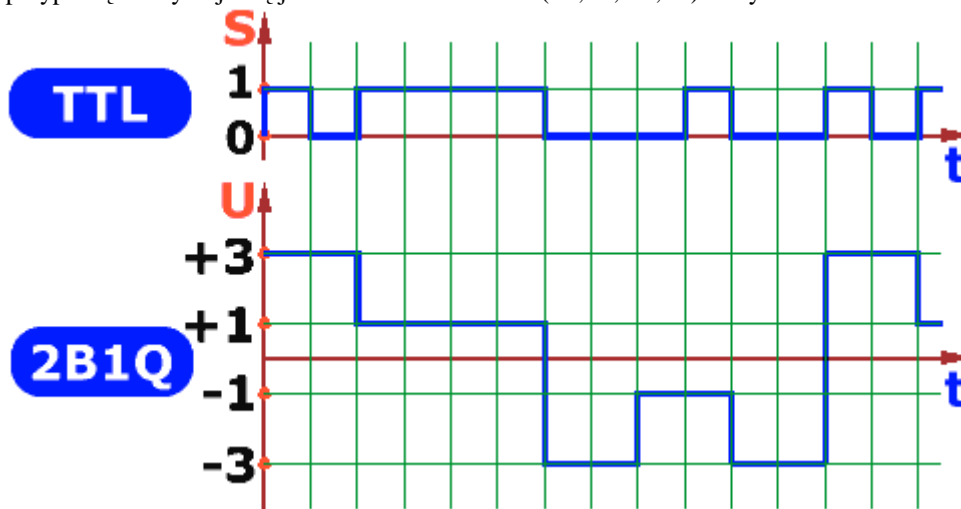


modem) a terminalem (styk S/T). Połączenie pomiędzy centralą a terminalem wykonane jest przewodem jednoparowym (styk U). Wykorzystuje się tam kodowanie 2B1Q



rysunek „2b1q.gif”

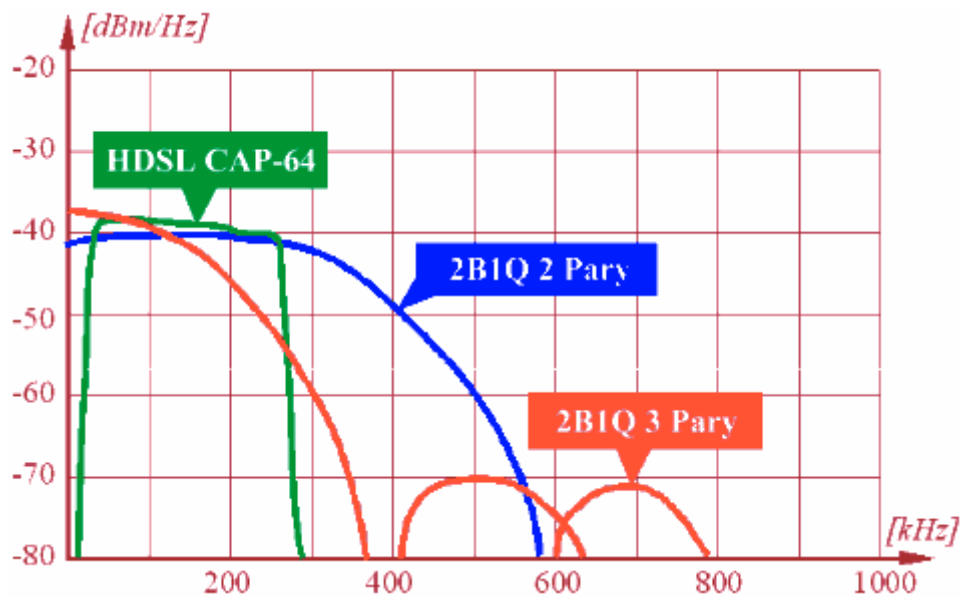
Polega ono na wyodrębnianiu z pakietu danych przeznaczonych do wysłania kolejno dwóch bitów, którym to przyporządkowuje się jeden z czterech stanów (+1, -1, +3, -3). Przykład kodowania ciągu bitów do 2B1Q



rysunek „2b1q-przyklad.gif”

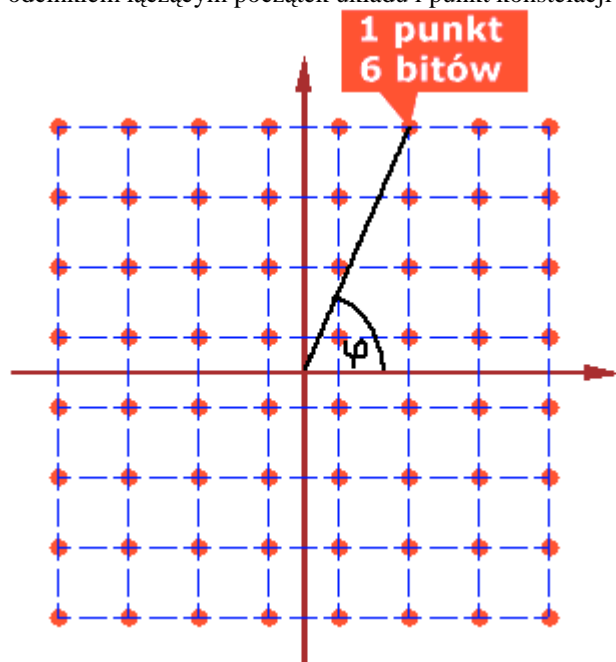
Skrót HDSL pochodzi od słów Hight-bit-rate Digital Subscriber Line, które w wolnym tłumaczeniu znaczą tyle, co Cyfrowa Linia Abonencka o Wysokiej Przepływności Bitowej. Czasem można się spotkać z rozwinięciem tego skrótu mającym postać Hight Digital Subscriber Loop, co rozumie się podobnie jak powyżej. HDSL korzysta ze starych łączy telefonicznych, nawet takich, co pamiętają jeszcze czasy Bell'a (XIX wiek). Fenomen tego rozwiązania polega na tym, że używa ono pasma znacznie szerszego niż 300 - 3400Hz, jakie jest stosowane do przenoszenia głosu, ale rzędu 6 - 259kHz. Własność ta wynika ściśle z Prawa Shannona. Początkowo technologia DSL korzystała z trzech par skrętki telefonicznej do przesłania 2Mb/s, lub jednej do przesłania 384kb/s (SDSL). Następnie pojawiły się na tyle skuteczne metody kodowania sygnału, które umożliwiły budowanie łączy 2Mb/s w technologii HDSL za pomocą dwóch par kabli telefonicznych. Ostatnio nawet realizuje się takie przepływności za pomocą jednej pary kabli.

W szybszych modemach wykorzystuje się inne sposoby kodowania niż 2B1Q. Charakterystyka widmowa różnych technik kodowania:



Rysunek „widma\_kodowanie.gif”

Tablica kodowania CAP64 wygląda jak na rysunku powyżej. Jest to konstelacja dwuwymiarowa. Składa się ona z 64 punktów charakterystycznych nie nakładających się na siebie, tak więc jednoznacznie rozpoznawalnych. 64 jest 6 (szósta) potęgą 2. Tak więc 6 jest liczbą bitów, jaką reprezentuje jeden punkt na tablicy. Modulacja przebiega w następująco: kolejnym 6 bitom jest przyporządkowywany jeden punkt konstelacji, którego opisem jest amplituda sygnału, czyli odległość punktu do środka układu oraz przesunięcie fazowe, czyli kąt między odcinkiem łączącym początek układu i punkt konstelacji z osią poziomą OX.



rysunek „cap\_64.gif”

W przypadku kodowania CAP 128 ilość punktów konstelacji wzrasta do 128 i na tym właściwie polega główna różnica między CAP 64, a CAP 128.

### Konfigurowanie modemu i programów

Aby modem działał poprawnie musimy skonfigurować program terminala lub inny o podobnym przeznaczeniu i modem. Bardzo często zadanie to wykonuje za nas instalator, ale gdy coś działa nieprawidłowo, chcemy poprawić

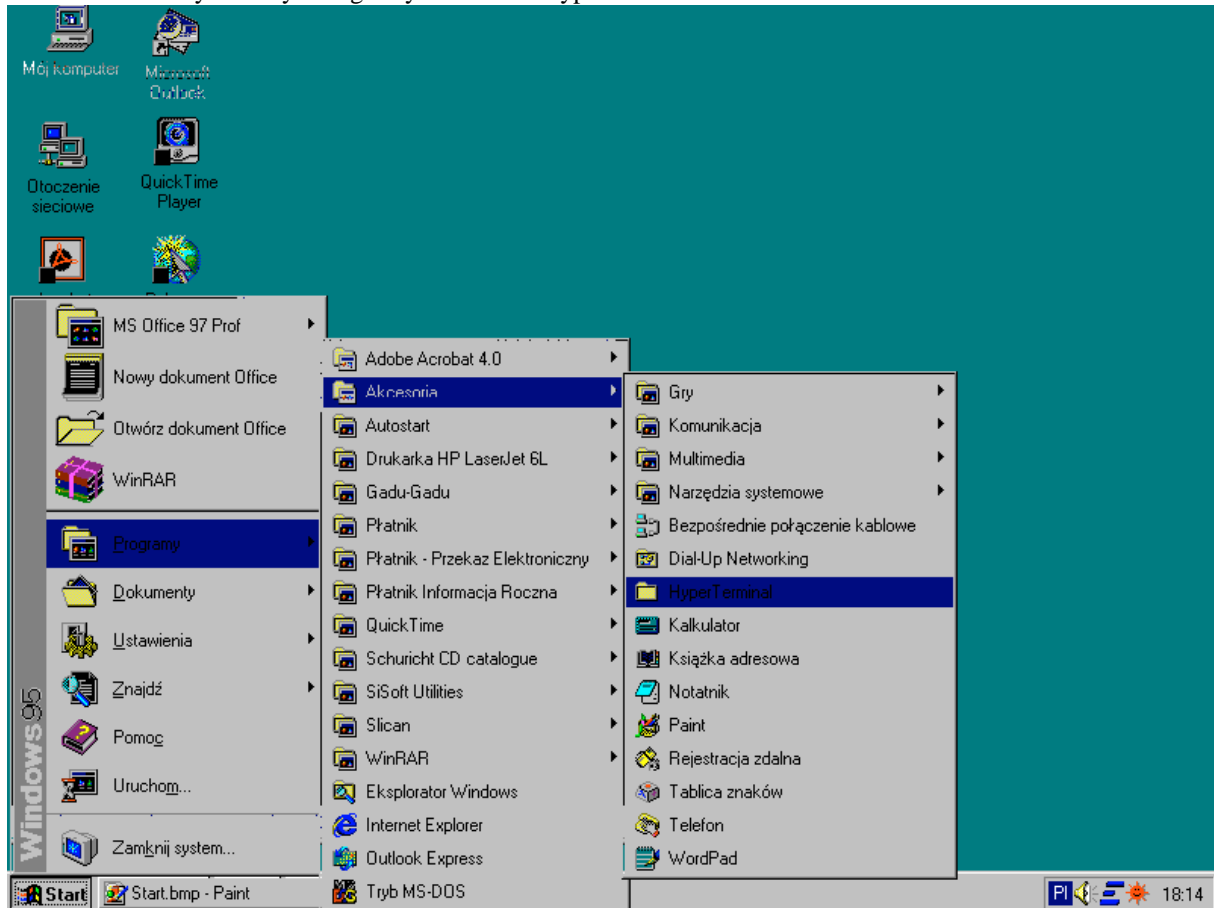
parametry transmisji lub po prostu poznać dokładniej modem warto zaznajomić się ze szczegółami związanymi z konfiguracją.

## **Konfigurowanie programu terminala**

Pierwszym krokiem będzie zapoznanie się z komendami AT. Zanim jednak to nastąpi opiszę konfigurację najpopularniejszych programów.

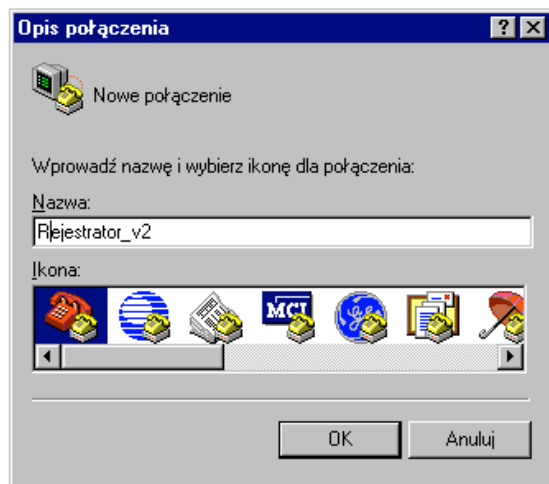
### **HyperTerm**

Z menu "Start" wybieramy "Programy/Akcesoria/HyperTerminal"



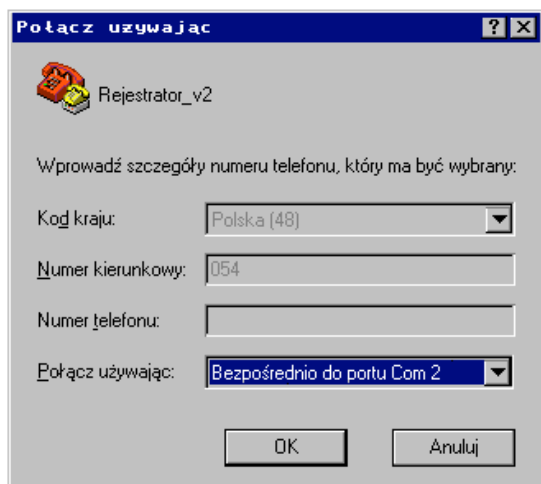
Rys "TermE1.pcx"

W oknie które pojawi się, klikamy na ikonce "HyperTerm.exe". Po uruchomieniu programu pojawi się oko



„TermE2.pcx”

Wpisujemy nazwę (np TestModemu), wybieramy ikonkę (domyślna jest OK) i zatwierdzamy przez naciśnięcie OK. W następnym oknie



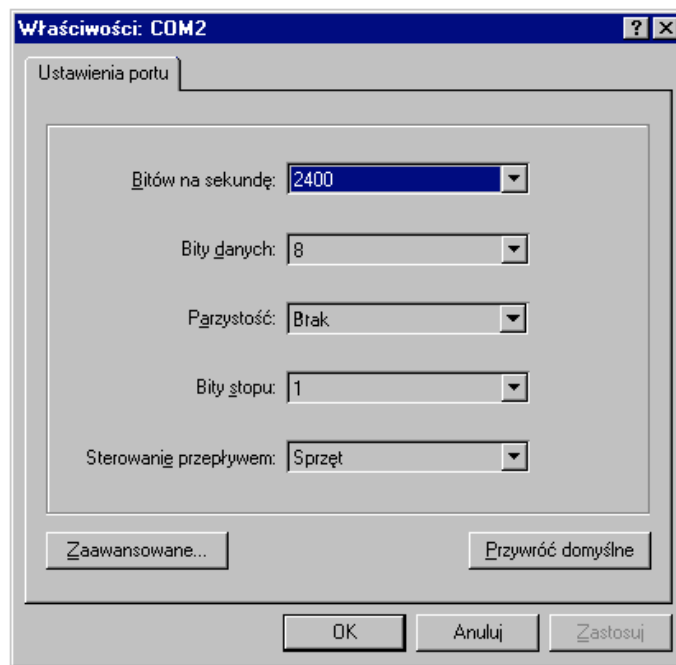
Ekran „TermE3.pcx”

przechodzimy od razu do listy "Połącz używając" i wybieramy "Bezpośrednio do COM2" (nr COM zależy od tego do jakiego portu komputera podłączyliśmy się). Zmiany zatwierdzamy naciskając OK. Jeśli posiadamy tzw. WinModem, modem na karcie PCI lub modem USB musimy koniecznie wybrać odpowiedni sterownik. W przypadku większości modemów analogowych można zamiast „Bezpośrednio do COM2” wybrać „Modem Standardowy”.

**UWAGA!**

W niektórych edycjach Windows nie można skonfigurować bezpośredniego dostępu do portu COM nie wypełniając wcześniej nr telefonu i innych zbędnych w takim przypadku rzeczy. Jedyne wyjście w takiej sytuacji to przebrnąć przez trudności lub zmienić system operacyjny.

W oknie "Właściwości" ustawiamy następujące parametry transmisji



Ekran „TermE4.pcx:

- Prędkość transmisji: 2400
- Bity Danych: 8
- Parzystość: Brak
- Bity stopu: 1
- Sterowanie przepływem: Sprzęt

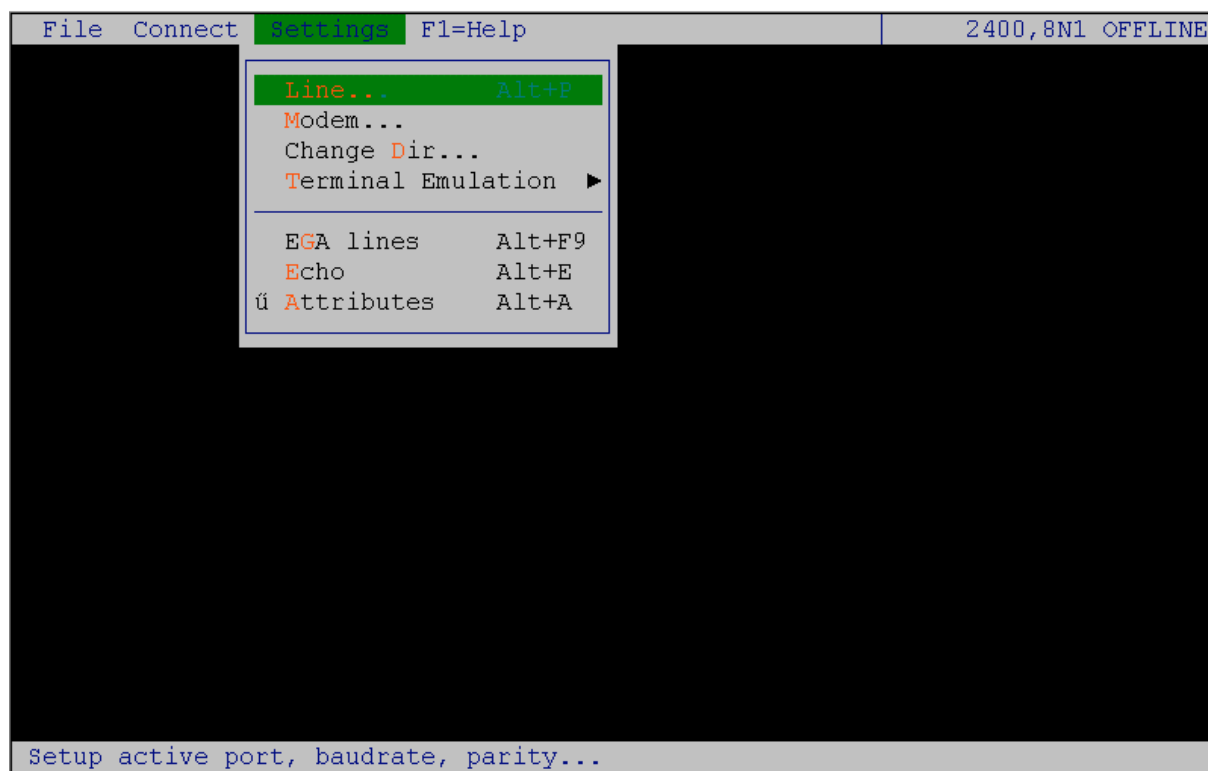
Oczywiście prędkość transmisji jest dość dowolna. Zmiany zatwierdzamy naciskając OK. Program jest już skonfigurowany i można go używać. Po wyjściu z programu można go uruchomić klikając na ikonkę "TestModemu.ht".

Mała uwaga dla użytkowników profesjonalnych wersji Windows: Blil G. Stwierdził, że wszystko wie najlepiej i nie pozwala skonfigurować HyperTerma od razu z ustawieniem „Bezpośrednio do portu Com xx”. Konieczne jest wpisanie nr telefonu i wiele innych niepotrzebnych czynności. Dopiero po takiej konfiguracji można wybrać ustawienia bezpośrednio do portu COM.

Kolejna rewelacja programów Microsoftu. To brak możliwości wybrania przez program dialera dla Windows znaku „!” , który generuje FLASH na linii. Funkcja taka jest potrzebna w niektórych centralach abonenckich, bez niej możemy odciąć się od możliwości obsługi wewnętrznych połączeń modemowych (przydatne przy zdalnej obsłudze centrali abonenckiej).

## **Term90**

Program "Term90" wchodzi w skład pakietu NortonCommander dla PC.  
W "Term90" w menu "Setings" wybieramy "Line.."

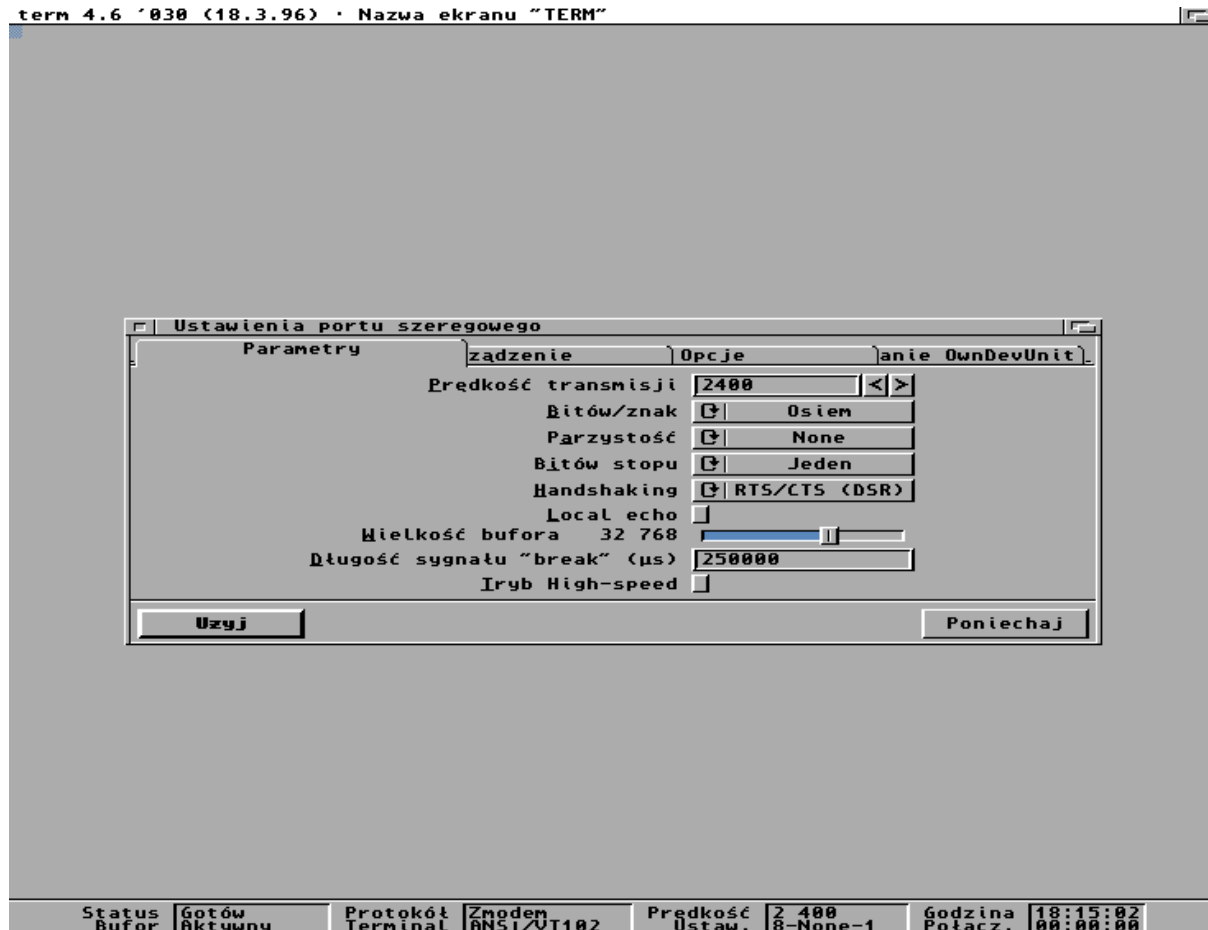


ekran „TermE5.pcx”

Ten sam efekt da naciśnięcie kombinacji klawiszy Alt+P. W oknie, które ukaże się, ustawiamy parametry transmisji (ekran TermE6.pcx).

#### **Term 4.6**

Program "Term" dla Amigi konfigurujemy wybierając w menu "Ustaw" podmenu "Port szeregowy". Opcje ustawiamy jak na zrzucie z ekranu



„ekran TermE7.pcx”  
akceptujemy naciskając "Użyj".

### !!!!!! Linux miniterm

W chwili pisania książki nie miałem dostępu do MAC'a i nie mogłbym zrobić zrzutów ekranowych. Konfiguracja programu dla tego komputera również jest banalnie prosta. Jak widać najbardziej pracochłonny i mało logiczny w konfiguracji jest program terminala pod Windows (gratulacje dla Microsoftu).

## Konfigurowanie modemu

W dalszej części książki opiszę sposób skonfigurowania modemu. Do pełnej konfiguracji potrzebna jest znajomość wielu komend AT, opisanych w kolejnych rozdziałach. Na szczęście do podstawowej konfiguracji wystarczy kilkanaście komend AT.

## Podstawowe rozkazy modemu analogowego

Uruchamiamy program terminala, wydajemy komendę "AT". Komendy będą ujmował w cudzysłów, każdą komendę kończymy naciśnięciem klawisza enter. W odpowiedzi na "AT" powinniśmy otrzymać komunikat "OK". Modem potwierdza każdą komendę, "OK" jeśli ją wykonał, "ERROR" jeśli nie (np. zły parametr). Następnie wydajemy komendę "ATZ" i łączymy się z innym modemem. Proponuje przed dalszymi próbami włączyć głośnik modemu tak aby było w nim słycać wybieranie nr i ustalanie parametrów transmisji. Po nawiązaniu połączenia głośnik wyłączy się. Robimy to komendą: „ATL3&M2”  
Nie każdy ma pod ręką drugi modem, ale to nie problem, można skorzystać z modemu TPSA. Wpisujemy w terminalu „ATDT0202122” jeśli central wybiera tonowo lub „ATDP0202122” jeśli impulsowo. Modem powinien nawiązać połączenie, przerywamy je wysyłając kod esc ("+++"). Kod znaku escape (+++) możemy zmienić w rejestrze S2. Po otrzymaniu komunikatu "OK" (czas od odebrania znaków +++ do wysłania OK.



ustawiamy w rejestrze S12) wysyłamy "ATH". Jeśli powyższa próba powiodła się, modem jest wstępnie ustawiony, ale możemy tam co nie co poprawić. Jeśli modem wogóle nie wybrał nr, ale zajął linię telefoniczną (najczęściej słychać „klapnięcie” przekaźnika) musimy zmienić ustawienia rozkazem „ATX”. Standardowe ustawienie (na 4) powoduje, że modem nie wybierze nr zanim nie rozpozna tonu zgłoszenia centrali 400Hz. Czas oczekiwania na ton można zmienić w rejestrze S6. Na centralach analogowych od momentu podniesienia słuchawki do momentu otrzymania tonu 400Hz mija od 0,5 do nawet kilku sekund. Po wydaniu komendy „ATX4” mamy gwarancję, że modem nie zacznie wybierać nr dopóki centrala nie wyśle tonu 400Hz dzięki czemu cały nr wybrany przez modem zostanie odebrany przez centralę. Gdyby wybieranie rozpoczęło się przed otrzymaniem tonu 400Hz początek nr mógłby zostać „zjedzony”. Po włączeniu usługi telekomunikacyjnej (np. przeniesienie nr) ton 400Hz z centrali nie jest ciągły lecz ma krótkie (około 500ms) przerwy. Uniemożliwia to niektórym modemom poprawne rozpoznanie sygnału zgłoszenia centrali. Podobna sytuacja może mieć miejsce w przypadku podłączenia modemu do centrali PABX. Rozkaz „ATX” umożliwia prace modemu w takich warunkach. Wystarczy wydać rozkaz „ATX0” i modem nie będzie czekał na ton 400Hz. Wskazane jest wtedy przed wydaniem komendy wybierania nr odczekanie kilku sekund lub komendę wybierania wydać w postaci „ATD,123”. Przecinek da pauzę przed nr dzięki czemu centrala powinna już rozpoznać nasze zgłoszenie (podniesienie słuchawki).

Jeśli połączenie nie zostało nawiązane lub zaraz po jego nawiązaniu przerwane, to musimy przywołać ustawienia fabryczne, wydając komendę "AT&F". Ponawiamy próbę połączenia, jeśli nie powiedzie się, próbujemy innych ustawień fabrycznych "AT&F1". Niektóre modemy oferują więcej ustawień fabrycznych np. "AT&F2". Szczerze mówiąc nie spotkałem modemu, który odmówiłby poprawnego działania po wydaniu komendy "AT&F0". Warto zaznaczyć, że "AT&F" i "AT&F0" znaczą to samo, po prostu modem jeśli nie podamy parametru komendy przyjmuje zero. Jeśli ustawienia fabryczne nie przyniosły spodziewanego skutku proszę spróbować wydać komendę: "AT&F&R0&K3&D2W2X4". Po jej wydaniu 99% modemów zaczyna działać.

Przed dalszymi "manewrami" zapisujemy aktualną konfigurację modemu "AT&W1". W razie kłopotów zawsze możemy ją przywrócić wydając komendę "ATZ1". Często standardowe ustawienia modemu powodują, że w komunikacji o połączeniu informowani jesteśmy o prędkości modem-komputer. Prędkość ta jest duża ale ma się nijak do faktycznej prędkości transmisji. Proponuję komendą „ATW2” ustawić sobie wyświetlanie prędkości modem-modem.

Jeśli modemem skonfigurowaliśmy tak jak chcemy, zapisujemy konfigurację komendą "AT&W". Teraz w programach, z których korzystamy, zmieniamy tzw. init stringi na "ATZ\r". "r" oznacza kod klawisza enter, w niektórych programach jest zastępowany sekwencją "M^". Jeśli mamy dwa rodzaje programów wymagających różnych konfiguracji, to drugą zapisujemy komendą "AT&W1". W init string takiego programu wpisujemy "ATZ1". Jeśli chcemy korzystać z większej liczby konfiguracji, nie pozostaje nam nic innego jak wpisać ją w initstring.

Przykładowa sprawdzona konfiguracja wygląda tak: "AT&F&R0&K3&D2W2X4"

## S-rejestry

Każdy modem posiada S-rejestry. Umożliwiają one bardziej zaawansowane konfigurowanie modemu. Aby zapisać dana do rejestru wydajemy komendę „ATSx=y” gdzie x nr rejestru, y wartość wpisywana do rejestru. Wszystkie dane wprowadzamy w systemie dziesiętnym. Aby odczytać rejestr należy wydać komendę „ATSx=?”, gdzie x nr rejestru. Po wydaniu komendy na ekranie terminala ujrzymy zawartość rejestru. Wynik zawsze jest trzycyfrowy, jeśli trzeba wartość jest poprzedzona zerami.

**Rejestr 0** odpowiada za to po ilu dzwonekch modem automatycznie odbierze połączenie. Jeśli wpisujemy 0 modem nie odbierze połączenia. Ustawienia tego rejestru mogą być mylące, ponieważ połączenie może odebrać program terminala komendą "ATA" po ustawionej w nim liczbie dzwonekch.

**Rejestr1** - licznik odebranych dzwonekch. Po każdym dzwonku licznik ten jest zwiększany o 1.

**Rejestr2** - kod znaku escape, standardowo 43 (znak "+"). Aby w czasie połączenia przejść do trybu rozkazowego należy wysłać trzy znaki esc. Po wysłaniu kodów escape należy poczekać na komunikat „OK”. Jeśli w tym czasie (czas ustawiamy w rejestrze S12) modem odbierze jakieś znaki uzna, że sekwencja esc znalazła się w transmitowanym pliku, nie wyśle komunikatu „OK.” i nie przejdzie do trybu rozkazowego. Takie postępowanie ma na celu uniknięcie przypadkowego przerwania transmisji. Rejestr 2 decyduje jaki to będzie znak.

**Rejestr3** - kod znaku CR, standardowo 13. Rejestr ten decyduje znak będzie traktowany przez modem jako kod powrotu karetki. Kod ten będzie także wysyłany przez modem do terminala gdy będzie on chciał przesunąć karetkę.

**Rejestr4** - kod znaku LF, standardowo 10. Rejestr ten decyduje znak będzie traktowany przez modem jako kod wysunięcia wiersza. Kod ten będzie także wysyłany przez modem do terminala gdy będzie on chciał przejść do nowej linii.

**Rejestr5** - kod znaku bakspace, standardowo 8. Rejestr ten decyduje znak będzie traktowany przez modem jako kod usunięcia znaku. Kod ten jest istotny gdy pracujemy w terminalu i pomylimy się. Po wysłaniu kodu bakspace ostatni znak zostanie usunięty z bufora modemu umożliwiając poprawienie pomyłki.

**Rejestr6** - czas oczekiwania na sygnał zgłoszenia centrali w sekundach, standardowo 5. Zależnie od ustawienia modemu rozkazem „ATX” modem przed wybraniem nr może oczekiwać na ton zgłoszenia centrali 400Hz. Jeśli w odpowiednim czasie ton nie pojawi się otrzymamy komunikat „NO DIAL TONE”

**Rejestr8** - czas trwania pauzy (znak ",") w sekundach, standardowo 2. Jeśli w rozkazie wybierania nr (ATD) użyjemy znaku pauzy wybieranie zostanie na chwilę przerwane. Opcja przydatna w centralach PABX gdzie po wybraniu 0 (wyjście na miasto) należy chwilę poczekać na sygnał. Dzięki pauzie mamy gwarancję, że część nr nie zostanie „zjedzona”

**Rejestr10** - opóźnienie od zmiany stanu linii DCD na nieaktywny do rozłączenia (zwolnienia linii telefonicznej) w 1/10 sek, standardowo 7 (0.7sek). Gdy modem wykryje brak nośnej zmienia stan linii DCD. W rejestrze tym decydujemy po jakim czasie ma to nastąpić. Ustawienie dłuższego czasu może uchronić nas przed rozłączeniem połączenia w przypadku chwilowej utraty fali nośnej.

**Rejestr11** - czas trwania tonu DTMF w ms (od 50 do 100), standardowo 70. Minimalny czas nadawania i czas przerwy tonów DTMF jest ściśle określony w normach telekomunikacyjnych. Im czas ten jest krótszy tym szybciej zostanie wybrany nr. Jeśli jednak czas ten będzie zbyt mały centrala może nie rozpoznać wybranej cyfry.

**Rejestr12** - czas od wysłania kodu ESC do otrzymania komunikatu "OK" w 1/50 sek. standardowo 75. Aby w czasie połączenia przejść do trybu rozkazowego należy wysłać trzy znaki esc (standardowo „+”, kod znaku można zmienić w rejestrze S2). Po wysłaniu kodów escape należy poczekać na komunikat „OK”. Jeśli w tym czasie modem odbierze jakieś znaki uzna, że sekwencja esc znalazła się w transmitowanym pliku, nie wyśle komunikatu „OK.” i nie przejdzie do trybu rozkazowego. Takie postępowanie ma na celu uniknięcie przypadkowego przerwania transmisji. Rejestr S12 decyduje o tym przez jaki czas po wysłaniu znaków esc modem nie może odbierać znaków aby przejść do trybu rozkazowego.

**Rejestr18** - Test timera, jeśli 0 to timer wyłączony, standardowo 0

**Rejestr25** - czas przez który modem ignoruje stan linii DTR w sek. standardowo 0. Jak było wspomniane wcześniej linią DTR możemy rozłączyć połączenie lub przejść do trybu rozkazowego. Rejestr 25 decyduje o tym po jakim czasie to nastąpi.

**Rejestr26** - czas opóźnienia od momentu zmiany stanu linii RTS na nieaktywny (wysłania znaku XOFF) do odpowiedzi na CTS w sek. standardowo 0. Zależnie od ustawienia modemu rozkazem „AT&R” i „AT&K,, stan linii CTS może odzwierciedlać stan RTS.

**Rejestr29** - czas trwania impulsu FLASH w 1/10sek, standardowo 20. Modem podczas wybierania nr może generować flash. Rejestr S29 decyduje jaki długi ma to być impuls. W polsce najczęściej flash zawiera się w granicach 500-700ms.

**Rejestr32** - kod znaku XON, standardowo 17. Gdy komendą „AT&K” ustawimy programowe sterowanie przepływem rejestr S32 możemy zdecydować jaki znak będzie wznawiał transmisję.

**Rejestr33** - kod znaku XOFF, standardowo 19. Gdy komendą „AT&K” ustawimy programowe sterowanie przepływem rejestr S33 możemy zdecydować jaki znak będzie powstrzymywał transmisję.

**Rejestr38** - czas od wysłania komendy "ATH" do rozłączenia (zwolnienia linii telefonicznej) w sek., standardowo 0

**Rejestr95** - odpowiada za komunikaty dotyczące protokołów połączeń. Jeśli wpisujemy tam wartość 255 uzyskamy najwięcej i najbardziej szczegółowych komunikatów.

Warto zaznaczyć, że wszystko co ustawiamy innymi komendami ma odzwierciedlenie w rejestrach, np. ustawiając bit 0 rejestru S14 włączamy echo. Oczywiście nie musi to dotyczyć wszystkich modeli modemów.

## Konfiguracja modemu ISDN na przykładzie modemu TA128



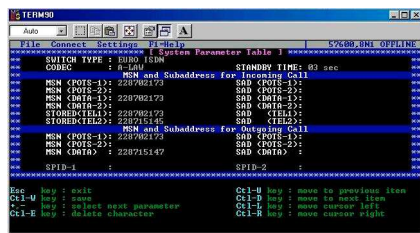
Fotografia „asmax\_isdn\_ta128.jpg”

Modem można skonfigurować wydając odpowiednie komendy AT. Można też skorzystać z programów konfiguracyjnych dostępnych w Internecie. Pierwszy sposób jest niezbyt wygodny, drugi zupełnie niepotrzebny ponieważ oprogramowanie modemu zawiera prosty i przejrzysty program konfiguracyjny. Po wydaniu komendy „At@” naszym oczom ukaże się ekran programu konfiguracyjnego:



## PLIK „TAE1.JPG”

Po naciśnięciu klawisza „2” zobaczymy okno:



## PLIK „TAE2.JPG”

Klawiszami kursora poruszamy się po oknach. Miejsce edycji wskazuje kursor. Znaczenie parametrów jest następujące:

- SWITCH TYPE** - rodzaj centrali ISDN wybieramy „EURO ISDN”
- CODEC** - rodzaj kodeka dla portów analogowych, wybieramy „A-LAW”
- STANDBY TIME** - okres czasu pomiędzy wybraniem ostatniej cyfry w porcie analogowym a wysłaniem żądania (wybraniem nr), najlepsze ustawienie 03-05 sec
- MSN (Incoming)** - w przypadku posiadania nr MSN wpisujemy na jakie nr mają reagować porty analogowe, a na jakie modem.
- MSN (Outcoming)** - jak wyżej tyle, że dla ruchu wychodzącego (jakim nr mają prezentować się porty analogowe, a jakim modem)
- SPID** - ten parametr ustawiamy tylko w Stanach Zjednoczonych, w pozostałych przypadkach bezwzględnie wyłączyć.
- SAD** - sub-adres

## Komendy Hayes AT

Rozdział ten opisuje komendy Hayes AT, uznawane za standardowe, ich składnię i rezultat zwracany przez komendę. Opisane zostaną też komunikaty generowane przez modem. Komendy AT możemy przetestować przy użyciu dowolnego komputera wyposażonego w port szeregowy lub USB (Amiga, AtariST, PC, MAC a nawet C-64 czy Atari) i modemu zewnętrznego. Możliwe są też testy na PC i modemie w postaci karty PCI lub ISA. W przypadku kart należy ostrożnie podchodzić do tzw. WinModemów, z którymi komunikacja nie zawsze przebiega przez port szeregowy (tak samo zresztą jak i z modemami USB) lecz przez odpowiedni sterownik. Z tego powodu programy, które odwołują się bezpośrednio do rejestrów portów szeregowych nie koniecznie będą działać (np. Term90). Jedyne wyjście jakie mamy w tej sytuacji to programy windosowe (np. Hyper Term), korzystające ze sterowników systemowych.

## Standardowe komendy AT

Po podłączeniu modemu do komputera i włączeniu komputera uruchamiamy program terminalowy. Następnie wpisujemy „AT” po czym naciskamy ENTER. W oknie terminala powinniśmy ujrzeć odpowiedź "OK,.". Jeśli w modemie włączone jest ECHO (włączamy je komendą „ATE1” [ENTER]) to wszystkie znaki które naciskamy na klawiaturze będziemy widzieć w oknie terminala. Jeśli po komendzie „AT” nie otrzymaliśmy odpowiedzi „OK” to oznacza, że wybraliśmy zły nr portu szeregowego, modem jest nie włączony lub nie podłączony. Po usunięciu usterki możemy wypróbować wszystkie komendy AT.

Przed opisaniem komend zaznaczę, że na różnych modemach niektóre komendy działają różnie. Aby być pewnym co dana komenda tak naprawdę robi należy przeczytać instrukcję obsługi danego modemu. Różne jest też zachowanie modemów podczas odsyłania echa. Większość z nich odsyła wszystkie znaki jakie wysyłamy do modemu. Spotkałem się jednak z takim modemem, który odsyłał znaki tylko gdy układały się one w komendę AT.

Na CD dołączonym do książki znajduje się spis wszystkich standardowych komend AT.

Aby uprościć opis komend pominąłem w nich początkowe znaki „AT”. Należy więc pamiętać, że wszystkie opisane poniżej komendy poprzedzamy znakami AT. Komendę zatwierdzamy naciskając klawisz enter (wysyłając znak CR o kodzie \$0D). Poniżej spis komend z komentarzami. Pogrubioną czcionką oznaczono komendę, a w przypadku gdy komenda ma parametry pogrubiona czcionka oznacza standardowe ustawienie.

**A/** powtórz ostatnią komendę. Po wydaniu tego rozkazu modem wykona ponownie poprzedni rozkaz. Gdy np. wydamy komendę wybierania nr (ATDT0123456789) i nie połączymy się, to aby nr wybrać ponownie nie musimy powtarzać tasiemkowej komendy wystarczy że wpisujemy „**ATA**”.

**A** odebranie połączenia. Gdy w oknie terminala ujrzymy komunikat „RING” a modem nie ma ustawionego samozgłoszenia (rejestr S0 ma wartość równa 0) po wydaniu rozkazu **ATA** modem odbierze połączenie i będzie próbował „dogadać się” z modemem po drugiej stronie drutów. Owo dogadanie się polega najczęściej na wynegocjowaniu jak największej prędkości transmisji oraz ustaleniu protokołu kompresji danych. Gdy modemu ustanowią połączenie zostaniemy zawiadomieni o tym komunikatem „CONNECT”.

**Bn** protokół poniżej 2400, gdzie n=[0..1]  
0 - CCITT v.21/v.22

#### **1 - Bell 103/212A**

Rozkazem tym wybieramy protokół komunikacyjny dla połączeń o prędkości mniejszej niż 2400bd. Przy dzisiejszych modemach i prędkościach transmisji rzędu dziesiątek a nawet (w przypadku ISDN) setek kbps (kilo bodów na sekundę) rozkaz ten można traktować jako ciekawostkę.

**Cn** Carrier control

#### **1 - Return OK message**

**D** wybieranie nr. Wydając komendę „ATD1234” spowodujemy wybranie numeru 1234. Sposób wybierania (tonowy czy impulsowy) będzie zależał od konfiguracji modemu. W komendzie poza cyframi można używać znaków specjalnych, oto ich znaczenie:

@ czeka 5 sekund

, pauza 2 sekundy (ustawianie w rejestrze S8)

! wygenerowanie flasha (czas flasha ustawiamy w rejestrze S29)

; po wybraniu nr powraca do interpretacji komend, nie próbuje nawiązać połączenia z drugim modemem. Parę słów wyjaśnienia do powyższej opcji. Może ona być przydatna gdy modem, z którym łączymy się jest podłączony do centrali abonenckiej i osiągalny po przez wybranie nr na tle zapowiedzi słownej (tzw DISA). Załóżmy, że centrala ma numer 1234567 modem nr wewnętrzny 123. Aby połączyć się z takim modemem wysyłamy komendę ATDP1234567; lub ATDT1234567; jeśli centrala wybiera tonowo. Po zgłoszeniu DISY (słuchać zapowiedź w głośniczku) wydajemy komendę ATDT123. Numer można wybierać także wydając komendę ATDnnnn. O tym czy wybieranie będzie tonowe czy impulsowe będzie decydować konfiguracja modemu.

#### **UWAGA!**

W Windowsowych programach Dialerów (Dial-Up Networking) osiągnięcie niektórych funkcji związanych z komendą ATD jest utrudnione lub wręcz niemożliwe. Jeśli są one potrzebne konieczna jest zmiana programu na inny (najlepiej odejść od jedynie słusznego systemu) lub wydanie komendy wybierającej nr „z palca” w programie terminala.

**P** wybieranie impulsowe. Wydanie rozkazu „ATDP123” spowoduje impulsowe wybranie nr 123.

**T** wybieranie tonowe. Wydanie rozkazu „ATDP123” spowoduje impulsowe wybranie nr 123.

Dwie powyższe opcje nie mają znaczenia w przypadku modemów ISDN, GSM czy SDI. Modemy te bez względu na ustawienia wybiorą nr w właściwy dla siebie sposób. W przypadku SDI nie następuje nawet wybieranie nr. SDI bez względu na wpisany nr modem łączy się z dostawcą usług internetowych.

**Sn** wybranie numeru z książki modemu (n=0...3). Komenda „ATDS0” powoduje wybranie nr z książki zapisanego pod pozycją 0. Nowsze modemy udostępniają książkę z kilkunastoma, a modemy GSM kilkudziesięcioma czy nawet kilkuset wpisami. Ponadto niektóre modemy (wszystkie modemy GSM) umożliwiają dopisanie do nr komentarza np. nazwiska. Modemy GSM umożliwiają przeszukiwanie książki na podstawie komentarzy.

To już ostatnia opcja do rozkazu „ATD”. Przechodzimy do kolejnych komed.

**L** powtórz ostatni nr

**En** echo komend, gdzie n=[0..1]  
0 - wyłączone

## 1 – włączone

Włączenie echa ułatwia posługiwanie się programem terminala. Dzięki temu widzimy znaki, które naciskamy na klawiaturze. Wyłączenie echa może być przydatne w programach, które sterują modemem. Dzięki temu program nie otrzymuje znaków, które wysyła, a tylko te, które są odpowiedziami od modemu. Czasem ułatwia to pisanie oprogramowania, ponieważ mamy mniej danych do analizy.

**H** rozłączenie połączenia (Hang-Up), przyłączenie do linii.

Rozkaz „ATH” lub „ATH0” rozłącza lub odrzuca połączenie. Jeśli rozkaz ATH zostanie wydany po nawiązaniu połączenia nastąpi jego rozłączenie. Niektóre modemy (najczęściej te z dekodermem CLIP, modemy GSM i ISDN) umożliwiają odrzucenie połączenia. W tym celu po odebraniu dzwonka wydajemy rozkaz „ATH”.

Abonent wywołujący usłyszy w słuchawce sygnał zajętości lub komunikat „Połączenie nie może być zrealizowane” a zależy od tego do jakiej centrali jest podłączony.

Rozkaz ATH1 przyłącza modem do linii. Rozkaz ten może być przydatny gdy łączymy się z modemem za pośrednictwem telefonistki. Wtedy to wydajemy komendę „ATD0123456;”. Po wybraniu nr podnosimy słuchawkę w telefonie podłączonym do modemu i wydajemy komendę „ATH1”. Gdy zgłosi się telefonistka prosimy o wybranie nr wewnętrznego. Następnie wydajemy komendę „ATH1”, modem po drugiej stronie zgłosi się i nastąpi nawiązanie połączenia.

**In** identyfikacja typu modemu, gdzie n=[0..6]

- 0 - kod produktu
- 1 - suma kontrolna ROM
- 2 - statystyka połączeń
- 3 - wersja ROMu
- 6 - typ modemu

Dzięki tej komendzie można zidentyfikować typ modemu, producenta, i wersje oprogramowania. Przydatne jest to w urządzeniach PnP. Jednak na małych procesorach nie starczyłoby pamięci programu na zapamiętanie danych wszystkich modemów a co dopiero na programy sterowników. Dlatego w małych urządzeniach stosuje się standardowe komendy AT a modem przed podłączeniem do urządzenia konfiguruje tak aby zachowywał się w standardowy sposób.

**Ln** poziom dźwięku głośnika, gdzie n=[0..3]

Modemy analogowe mają wbudowany głośnik. Dzięki niemu można słyszeć proces wybierania, uzgadnianie parametrów transmisji czy sama transmisję. W tanich modemach głośnik zastąpiony jest marnej jakości blaszką piezoelektryczną. W modemach tych głośność dźwięku można regulować właśnie komendą „ATL” w granicach od 0 (dźwięki najcichsze) do 3 (dźwięki najgłośniejsze). Są modemy, w których głośność regulujemy potencjometrem. Oczywiście jest, że w tym przypadku modem nie interpretuje komendy ATL. Tak samo jest w przypadku modemów ISDN oraz SDI gdzie właściwie negocjacja prędkości nie ma miejsca. W przypadku modemów GSM rozkaz „ATL” bardzo często reguluje głośność dźwięku w słuchawce, którą do takiego modemu można podłączyć. Wtedy to parametr rozkazu zawiera się w szerszych granicach np. 0..63.

**Mn** praca głośnika, gdzie n=[0..2]

- 0 - wyłączony
- 1 - wyłączony po nawiązaniu połączenia**
- 2 - głośnik zawsze włączony

Jeśli modem ma głośnik można zdecydować kiedy ma być włączony. Najczęstsze ustawienie to 1 (głośnik wyłączony po nawiązaniu połączenia). Dzięki temu słyszymy proces wybierania nr i możemy określić przyczynę „nie dogadania się” modemów. Pozatym jest to jedyna możliwość wybrania nr na tle zapowiedzi (DISY).

**Nn** dozwolona prędkość transmisji, gdzie n=[0..1]

- 0 - ustalona w rejestrze S37

### 1 - maksymalna możliwa

Rozkazem tym ustawiamy czy prędkość transmisji pomiędzy modemami ma być negocjowana czy z góry ustalona.

**Qn** kody zdarzeń, gdzie n=[0..2]

### 0 - włączone

- 1 - wyłączone
- 2 - włączone podczas inicjalizacji połączenia, wyłączone gdy odbiera

Komenda ta umożliwia wyłączenie komunikatów. Wtedy to nie pojawi się komunikat o dzwonku czy połączeniu. Praca z takimi ustawieniami jest możliwa ponieważ czy nastąpiło połączenie, czy jest dzwonek, czy połączenie zostało zerwane można dowiedzieć się badając stan odpowiednich linii modemu.

**P** wybieranie impulsowe

Wydanie rozkazu „ATP” ustawia impulsowy sposób wybierania. Rozkaz nie ma znaczenia dla modemów ISDN, SDI, GSM.

**Sn=x** zapis rejestru specjalnego "n" wartością "x"

Rejestry specjalne będą opisane w kolejnych rozdziałach.

**Sn=?** wyświetla zawartość rejestru x (zawsze trzycyfrowo)

**T** wybieranie tonowe

Wydanie rozkazu „ATT” ustawia tonowy sposób wybierania. Rozkaz nie ma znaczenia dla modemów ISDN, SDI, GSM.

**Vn** rodzaj kodów wynikowych, gdzie n=[0..1]

0 - numeryczna

**1 - numeryczna i tekstowa**

Komenda ta wybiera w jaki sposób modem będzie informował nas o błędach, dzwonieniu, itp. Przy ręcznej obsłudze korzystniejsze są komunikaty tekstowe, jednak dla programu obsługującego modem łatwiej jest analizować komunikat na podstawie zwróconej liczby. Spis kodów wynikowych znajduje się w kolejnych rozdziałach.

**Wn** informacja o szybkości, gdzie n=[0..2]

**0 - prędkość modem-komputer**

1 - j/w + rodzaj korekcji błędów

2 - prędkość modem-modem

Przy tej komendzie zatrzymamy się na chwilę. Często cieszymy się, że program pokazuje prędkość połączenia równą np. 57600, a połączenie odbywa się na niższej prędkości. Za ten stan jest odpowiedzialny powyższy rejestr. Jeśli ustawimy go na 2 będziemy wiedzieli z jaką faktycznie prędkością modem połączył się.

**Xn** rodzaj dzwonienia (wybierania), gdzie n=[0..4]

0 - nie czeka na 400Hz, ignoruje zajętość, daje komunikat CONNECT po nawiązaniu

połączenia

1 - j/w + prędkość połączenia

2 - czeka na 400Hz, ignoruje zajętość, CONNECT

3 - nie czeka na 400Hz, daje BUSY gdy zajętość, oraz CONNECT po nawiązaniu połączenia

**4 - czeka na 400Hz, daje CONNECT, BUSY lub NO DIAL gdy brak sygnału z centrali**

**przez 5 sekund**

Standardowe ustawienie (na 4) powoduje, że modem nie wybierze nr zanim nie rozpozna tonu zgłoszenia centrali 400Hz. Czas oczekiwania na ton można zmienić w rejestrze S6. Na centralach analogowych od momentu podniesienia słuchawki do momentu otrzymania tonu 400Hz mija od 0,5 do nawet kilku sekund. Po wydaniu komendy „ATX4” mamy gwarancję, że modem nie zacznie wybierać nr dopóki centrala nie wyśle tonu 400Hz dzięki czemu cały nr wybrany przez modem zostanie odebrany przez centralę. Gdyby wybieranie rozpoczęło się przed otrzymaniem tonu 400Hz początek nr mógłby zostać „zjedzony”. Po włączeniu usługi telekomunikacyjnej (np. przeniesienie nr) ton 400Hz z centrali nie jest ciągły lecz ma krótkie (około 500ms) przerwy. Uniemożliwia to niektórym modemom poprawne rozpoznanie sygnału zgłoszenia centrali. Podobna sytuacja może mieć miejsce w przypadku podłączenia modemu do centrali PABX. Rozkaz „ATX” umożliwia prace modemu w takich warunkach. Wystarczy wydać rozkaz „ATX0” i modem nie będzie czekał na ton 400Hz. Wskazane jest wtedy wydaniem komendy wybierania nr odczekanie kilku sekund lub komendę wybierania wydać w postaci „ATD,123”. Przecinek da pauzę przed nr dzięki czemu centrala powinna już rozpoznać nasze zgłoszenie (podniesienie słuchawki).

**Yn** reakcja po odebraniu określonej liczby spacji, gdzie n=[0..1]

**0 - brak reakcji**

1 - przerwanie połączenia

Odpowiednie ustawienie umożliwi alternatywny (do znaków „+++” o czym dalej) sposób przerywania połączenia. Liczbę spacji ustawiamy w S-rejestrach.

**Zn** przywrócenie zapisanej konfiguracji

0 - przywrócenie profile0 (zapisane wcześniej komendą &W0)

1 - przywrócenie profile1 (zapisane wcześniej komendą &W1)

Po skonfigurowaniu modemu ustawienia można zapisać. Najczęściej są dostępne dwa miejsca do zapisania konfiguracji.

**&Cn** sterowanie DCD, gdzie n=[0..1]

0 - zawsze 1

**1 - 1 po uzyskaniu połączenia**

Dzięki ustawieniu w opcji wartości 1 śledząc stan linii DCD może określić kiedy nastąpiło połączenie (DCD aktywne) oraz zerwanie transmisji (DCD nieaktywne)

**&Dn** reakcja na DTR, gdzie n=[0..2]

0 - brak reakcji

1 - przyjmowanie komend gdy DTR=0

**2 - gdy DTR przyjmie 0: przerwanie połączenia, przyjmowanie komend, włączenie auto**

**answer**

Przy standardowym połączeniu w łatwy sposób można uzyskać przerwanie połączenia ustawiając linię DTR w stan nieaktywny. Natomiast po wydaniu rozkazu „AT&D1” mamy łatwy sposób przechodzenia pomiędzy trybem rozkazowym, a transmisją danych.

**&Fn** przywołanie konfiguracji z ROMu, gdzie n=[0..1]

Rozkaz przydatny, gdy po naszych doświadczeniach modem nie chce poprawnie zestawiać połączeń. Wydanie tego rozkazu ustawia konfigurację, zalecaną przez producenta. Nie jest ona może optymalna, ale nie spotkałem jeszcze modemu, który nie działałby po wydaniu rozkazu AT&F. Niektóre modemy posiadają więcej profili (np. dla łączy dzierżawionych).

**&Kn** rodzaj potwierdzenia, gdzie n=[0..5]

0 - wyłączone

**3 - RTS/CTS**

4 - programowe Xon/Xoff

5 - transparent Xon/Xoff

Komenda ta wybiera w jaki sposób modem będzie zawiadamiał procesor o możliwości odbierania od niego znaków i na odwrót. Szczegółowy sposób sterowania liniami RTS/CTS będzie opisany w kolejnych rozdziałach.

**&Qn** (zalecane 5)

**&Pn** parametry wybierania impulsowego (czas zwarcia/czas przerwy), gdzie n=[0..3]

0 - 39%/61% 10Hz

**1 - 33%/67% 10Hz**

2 - 39%/61% 20Hz

3 - 33%/67% 20Hz

Obowiązującym w Polsce standardem jest 33%/67% 10Hz i nie mamy tu innego wyboru.

**&Rn** synchronizacja RTS/CTS, gdzie n=[0..1]

**0 - CTS reaguje na RTS**

1 - CTS=1

Komendą tą wybieramy Czy stan linii CTS ma odzwierciedlać stan linii RTS terminala, czy ma zawsze być ustawiony na 1 gdy modem ma włączone zasilanie.

**&Sn** kontrola DSR (zalecane n= 0)

**&Vn** wyświetlenie konfiguracji, gdzie n=[0..2]

0 - aktualna konfiguracja

1 - konfiguracja profile0 (zapisane wcześniej komendą &W0)

2 - konfiguracja profile1 (zapisane wcześniej komendą &W1)

Niektóre modemy po komendzie AT&V wyświetlają wszystkie konfiguracje. Warto przed testami modemu wydrukować sobie aktualną konfigurację. Dzięki temu łatwiej wrócić do ustawień przy których modem działał. Ja osobiście drukuję sobie konfiguracje modemu dobrą dla danego typu połączeń (internetowe, transmisja danych do innego modemu). Dzięki temu łatwiej jest skonfigurować kolejne modemy.

**&Wn** Zapisanie aktualnej konfiguracji w NVRAM, gdzie n=[0..1]

Komenda zapisuje aktualną konfigurację modemu do pamięci NVRAM (w nowych modemach w FLASH).

Dostępne są najczęściej dwie pamięci konfiguracji (np. połączenia internetowe i połączenia transmisji danych).

Zapisane wcześniej konfiguracje można przywrócić komendą ATZ lub ATZ1.

**&Xn** zegar synchronizujący, gdzie n=[0..1]

**0 - wewnętrzny**

1 - zewnętrzny

**&Yn** konfiguracja po włączeniu, gdzie n=[0..1]

**0 - profile0**

1 - profile1

Rozkazem tym decydujemy, która konfiguracja ma być wczytana po włączeniu zasilania (ta zapisana przez „AT&W0” czy „AT&W1”).

**&Zn=x** zapisanie numeru w książce telefonicznej modemu, gdzie x=[0..3]

Wydając komendę „AT&Z0=1234” pod pozycją 0 książki zostanie zapamiętany nr 1234. Nr ten możemy

wybrać wydając komendę „ATDS0”. Książkę telefoniczną najczęściej można obejrzeć na końcu konfiguracji (po wydaniu rozkazu „AT&V”).

**%Cn** kompresja, gdzie n=[0..3]

**0 - wyłączona**

1 - włączona MNP

2 - włączona V.42

3 - V.42 i MNP

Jeśli wykorzystujemy modem do własnych celów, polecam ustawić opcję na 0. W przeciwnym wypadku po "dogadaniu się" modemów i wysłaniu komunikatu "CONNECT" nastąpi ustalanie protokołu kompresji, co może objawić się wysłaniem sporej ilości "krzaczków".

**%En** reakcja na zmianę parametrów linii, gdzie n=[0..2]  
0 - brak reakcji  
1 - ponowna negocjacja prędkości zaczynając od 4800  
**2 - j/w z próbami zwiększania/zmniejszania prędkości**

Najlepsze ustawienie to oczywiście 2. Dzięki temu po zmianie parametrów linii na gorsze modemy nie rozłączają się czy też nie będą wiecznie czekały na przesłanie danych przy lepszych parametrach lecz na nowo ustalą największą możliwą prędkość transmisji. Dzięki temu transmisja będzie odbywać się dalej choć wolniej.

**%Gn** j/w dla V32.bis  
**\An** maksymalna wielkość bloku przy kompresji MNP  
0 - 64  
1 - 128  
2 - 192  
3 - 256

Najkorzystniejsze jest ustawienie największego możliwego bloku wtedy to dodatkowe dane (nagłówki, sumy kontrolne) zajmują procentowo mniej danych. Jeśli jednak zdarzają się częste powtórzenia bloków (błędy CRC) należy go zmniejszyć. Dzięki temu powtarzane są mniejsze porcje danych. W konsekwencji transmisja jest szybsza.

**\Nn** wybór protokołu (zalecane n=0)

Warto zaznaczyć, że to nie wszystkie komendy. Jedne modemy mają ich więcej, inne mniej. O szczegółach można poczytać w instrukcji obsługi. Dostępne rejestry i konfigurację możemy zobaczyć po wydaniu komendy AT&V. Ponadto niektóre komendy mogą mieć więcej parametrów, czasem mniej. Na niektórych typach modemu zdarzyło mi się, że po wpisaniu nieodpowiedniego parametru modem zawieszał się (czyżby był tam Win?). Najczęściej jednak wyświetlany jest napis ERROR. Czasem od modemu można uzyskać informacje jakie parametry akceptuje, wydając komendę w postaci komenda=? np.: AT&Y=?

W jednym rozkazie można wydać kilkanaście komend. Kolejne komendy łączymy znakiem „&”, natomiast AT piszemy tylko raz, np.: **AT&F&R0&K3&D2W2X4**

## Kody wynikowe

Komunikaty (liczbowe i tekstowe) zwracane przez modem. Komendą **ATV** decydujemy w jaki sposób modem będzie zwracał nam komunikaty. Komenda **ATV1** wybiera komunikaty tekstowe, **ATV0** numeryczne. Pierwsza opcja jest wygodna podczas pracy z modemem za pośrednictwem programu terminala, druga ułatwia analizę odebranych kodów przez mikrokontroler.

0 OK.	nie wymaga komentarza
1 RING	odebranie sygnału dzwonka
3 NO CARRIER	połączenie przerwano
4 ERROR	błąd
5 CONNECT 1200	połączenie z prędkością 1200bd
6 NO DIALTONE	brak sygnału zgłoszenia centrali
7 BUSY	abonent wybrany zajęty
8 NO ANSWER	nie doszło do połączenia
9 CONNECT 0600	połączenie z prędkością 600bd
10 CONNECT 2400	połączenie z prędkością 2400bd
11 CONNECT 4800	połączenie z prędkością 4800bd
12 CONNECT 9600	połączenie z prędkością 9600bd
13 CONNECT 7200	połączenie z prędkością 7200bd
14 CONNECT 12000	połączenie z prędkością 12kdb
15 CONNECT 14400	połączenie z prędkością 14,4kdb
16 CONNECT 19200	połączenie z prędkością 19,2kdb
17 CONNECT 38400	połączenie z prędkością 28,4kdb
18 CONNECT 57600	połączenie z prędkością 57,6kdb
19 CONNECT 115200	połączenie z prędkością 115,2kdb
22 CONNECT 75TX/1200RX	połączenie nadawanie 75bd odbiór 1200bd
23 CONNECT 1200TX/75RX	połączenie nadawanie 1200bd odbiór 75bd
24 DELAYED	opóźnienie
33 FAX	połączenie faksowe
35 DATA	połączenie transmisji danych



61 CONNECT 16800	połączenie z prędkością 16,8kba
62 CONNECT 21600	połączenie z prędkością 21,6kba
63 CONNECT 24000	połączenie z prędkością 24kba
64 CONNECT 26400	połączenie z prędkością 26,4kba
65 CONNECT 28800	połączenie z prędkością 28,8kba
66 CONNECT 33600	połączenie z prędkością 33,6kba
67 COMPRESSION: CLASS 5	kompresja CLASS 5
68 COMPRESSION: V.42 bis	kompresja V.42 bis
69 COMPRESSION: NONE	brak kompresji
70 PROTOCOL: NONE	brak protokołu
77 PROTOCOL: LAPM	protokołu LAPM
80 PROTOCOL: ALT	protokołu ALT
81 PROTOCOL: ALT-CELLULAR	protokołu ALT-CELLULAR

## Nawiązanie połączenia

Wydając komendę „ATDnnn”, gdzie nnn nr żądany, łączy się z drugim modemem (np. internet). Jeśli połączenie dojdzie do skutku, otrzymamy komunikat „CONNECT xxxx” gdzie xxxx prędkość transmisji. Dodatkowo linia DCD zmieni stan na aktywny. Po połączeniu, modem stanie się „przezroczysty” dla przesyłanych znaków. Połączenie rozłączamy komendą „ATH”, ale jak ją wydać skoro modem jest „przezroczysty” i nie interpretuje komend? W tryb rozkazowy wchodzimy naciskając trzy razy znak plus, po czym czekamy na komunikat OK., który pojawi się po około 1,5 sekundy. Od tego momentu jesteśmy w trybie rozkazowym i możemy wydać komendę „ATH”. Jeśli chcemy wrócić z powrotem do transmisji, wydajemy komendę ATO. Po niej dostaniemy komunikat „CONNECT”.

## Odbieranie połączeń

Połączenie przychodzące jest sygnalizowane komunikatem „RING”. W momencie kiedy modem odbiera dzwonek linia RI przyjmuje stan aktywny. Połączenie odbieramy wydając komendę „ATA”. Gdy dojdzie do połączenia otrzymamy komunikat „CONNECT”. Dodatkowo linia DCD zmieni stan na aktywny. Niektóre modemy posiadają dekodery CLIP’u. O standardach i formatach ramek CLIP można dowiedzieć się więcej w kolejnych rozdziałach. W DipModemie formy MultiTech do jego obsługi służą dwie komendy: „AT+VCID” i „AT+VCD”T. Komendą „AT+VCID” uaktywniamy bądź dezaktywujemy CLIP. „AT+VCID=0” wyłącza CLIP’a, „AT+VCID=1” włącza formatowany CLIP’a natomiast „AT+VCID=2” włącza CLIP’a w standardzie RAW. W tym przypadku modem nie ingeruje w przysyłąną informację i wyświetla ją bezpośrednio na ekranie. Swoją drogą przy CLIP’ie RAW można czasem wyczytać ciekawe informacje o centrali. Polecam jednak formatowany CLIP. W takim przypadku jeśli modem odbierze dzwonek na ekranie terminala zobaczymy:

RING

CIDM

DATE = 0614

TIME = 1014

NMBR = 0123456789

Znaczenie poszczególnych pól jest chyba jasne, nadmienię tylko, że: CIDM to informacja, że rozpoznano CLIP, DATE wyświetla datę w formacie MM/DD, TIME wyświetla czas w formacie gg/mm.

**Rozkazem AT+VCDT** wybieramy rodzaj CLIP’a.

AT+VCDT=0 wybiera CLIP FSK dekodowany po dzwonku. Taki CLIP obowiązuje oficjalnie od 03.2003 w TPSA (w praktyce bywa jednak różnie). Jeśli jednak mamy łącze ISDN, to możemy w terminalu wybrać standard DTMF.

AT+VCDT=1 wybiera CLIP FSK z tym, że oczekuje na CLIP zawsze.

AT+VCDT=2 wybiera standard UK

AT+VCDT=3 wybiera standard Japan

AT+VCDT=4 wybiera CLIP w standardzie DTMF. W tym standardzie nie mamy jednak informacji o dacie i godzinie. Informacja ta jest czasem przydatna, bo może synchronizować wewnętrzny zegar RTC. Reasumując, aby odbierać CLIP FSK należy wydać komendy:

AT+VCID=1

AT+VCDT=0

Jeśli CLIP jest dostępny w DTMF wydajemy komendy:

AT+VCID=1

AT+VCDT=4

Niestety modem nie ma komendy zapamiętującej konfigurację w NV-RAM (AT+&W) i po każdym resecie należy modem ponownie skonfigurować. Dla urządzeń, w które wbudowano modem nie stanowi to żadnej niedogodności. Procedury z ROM mogą wpisać do modemu wymaganą konfigurację po resecie. Do programowego zerowania modemu służy komenda ATZ. Po jej wykonaniu modem znajdzie się w takim stanie jak po sprzętowym resecie.

## Standardy i formaty ramek CLIP

Na łączach analogowych możliwe jest wysyłanie do abonenta informacji o nr abonenta wywołującego go. Istnieją dwa standardy informacji CLIP: DTMF i FSK. Na łamach EP pojawił się kilka rozwiązań dekodowników clip. W artykułach jednak po macoszemu omówiono standardy przesyłanych ramek skupiając się tylko na podstawowych ramkach (w przypadku FSK daty/godziny i nr abonenta wywołującego). Warto poznać dokładniej typy przesyłanych. Wiele central (Siemens, Slican) generuje dodatkowe ramki, a np. rejestrator telefoniczny AVT-50xx przesyła ramkę w formacie RAW na port szeregowy umożliwiając jej zdekodowanie. Ponadto niektóre telefony i identyfikatory wyświetlają dodatkowe ramki (np. opis tekstowy).

Na początku omówię prostszy **standard DTMF (Dual Tone Multi Frequency)**:

Spotkałem się z dwoma sposobami wysyłania kodów CLIP:

Kody DTMF (80ms ton, 80ms przerwa)	Pauza min 250ms max 1sek.	Dzwonek
------------------------------------	---------------------------	---------

Oraz:

Dzwonek (najczęściej krótki około 500ms)	Pauza min 250ms max 1sek.	Kody DTMF (80ms ton, 80ms przerwa)	Pauza min 200ms	Dzwonek
--	---------------------------	------------------------------------	-----------------	---------

W Polsce spotyka się pierwszą z omówionych tu metod. Kody DTMF układają się w ramkę złożoną z trzech pól:

Znak startu ramki	nr abonenta	znak końca ramki
-------------------	-------------	------------------

Numer abonenta może być 20cyfrowy, przykładowa ramka może wyglądać tak:

**D012345678A**

Powyższa ramka jest prawdziwa dla centrali ESS5. Mając do czynienia z różnymi typami central spotkałem się z różnymi kodami znaków startu i końca ramki. Np. dla SIEMENS'a ramka wyglądała tak:

**D0123456789C**

Na szczęście jest pewna reguła - znak startu i końca jest różny od 0...9, najczęściej są to kody A, C, D. Ramka abonenta zastrzeżonego lub bez możliwości identyfikacji w miejscu nr ma dziewięć lub dziesięć zer (zależne od modelu centrali).

Aby program dekodujący clip działał prawidłowo należało by postępować wg poniższego algorytmu:

1. Czekaj na znak
2. Jeśli odebrano znak z zakresu 0..9 lub # czy \* skocz do 1
3. Jeśli znak z zakresu 0..9 zapisz do bufora, zwiększ wskaźnik w buforze i skocz do 3
4. Jeśli w ciągu 3 sek. Nie ma dzwonka skocz do 1
5. Jeśli wszystkie odebrane znaki w buforze są zerami to nie ma możliwości zidentyfikowania rozmówcy
6. Znaki w buforze różne od zera – wyświetl je

Powyższy algorytm zapewnia dekodowanie CLIP'a na każdej centrali bez względu na sposób przesyłania CLIP'a i znaki startu oraz stopu. Naturalnie procedurę oczekiwania na znak należy zaopatrzyć w timeout aby program nie zawiesił się.

Jak łatwo zauważyć standard DTMF jest stosunkowo prosty, ale przekazuje mało informacji (nie można odróżnić abonenta z nr zastrzeżonym od abonenta bez możliwości identyfikacji). Ponadto wysłanie tylko 10 cyfr będzie trwało  $12(\text{łącznie ze znakami Start i Stop}) \times 160\text{ms}(\text{czas tonu} + \text{czas przerwy}) = 1920\text{ms}$  czyli prawie 2 sekundy! Przy 20 cyfrach będą to prawie 4 sekundy!

Znacznie bardziej zaawansowany i mający więcej możliwości jest **standard FSK**:

I tu są dwie metody wysyłania CLIP'a.. Standard **BELLCORE** (stosowany w Polsce):

Dzwonek (najczęściej krótki około 500ms)	Pauza min 250ms	Transmisja FSK od 500ms do 200ms	Pauza min 200ms	Dzwonek (standardowa długość 1sek.)
--	-----------------	-------------------------------------	-----------------	---

Metoda druga **BT** (British Telecommunications):

Zmiana polaryzacji żył	Pauza min 100ms	Sygnał synchronizacji	Pauza min 45 max 75ms	Transmisja FSK od 500ms do 200ms	Pauza min 200ms	Dzwonek (standardowa długość 1sek.)
------------------------------	--------------------	--------------------------	--------------------------	---	--------------------	---

W standardzie FSK jedynce logicznej odpowiada ton 1300Hz, zeru logicznemu 2100Hz (standard V.23),  
prędkość transmisji wynosi 1200bd. Kompletna ramka składa się z:

Synchronizacji – sygnał <b>SMRR</b> 80-250ms	Bitów stopu – sygnał <b>MARK</b> 55-160ms	Meldunku ( <b>MESSAGE</b> ) 500-2000ms
---	--	---

Pierwsze dane ramki (znak SMRR) składają się z bitów, których wartość zmienia się naprzemiennie  
0,1,0,1...Sygnał ten jest używany do zsynchronizowania dekodera FSK. Sygnał MARK złożony jest ze 180  
bitów o wartości 1. Po nim wysyłany jest meldunek.

Budowa meldunku jest następująca:

Bajt określający rodzaj danych <b>TYPE</b>	Bajt określający liczbę bajtów meldunku <b>LEN</b> (zależna od długości rekordów meldunku)	Rekord/rekordy meldunku	Bajt sumy kontrolnej <b>CRC</b> (suma modulo 2)
---	---	-------------------------	---

Pole TYPE najczęściej zawiera wartość \$80 – identyfikacja dzwoniącego.

Rekord meldunku ma następującą budowę:

Bajt rodzaju danych ( <b>BYPE</b> )	Bajt określający liczbę bajtów danych ( <b>LEN</b> )	Dane ( <b>DATA</b> )
-------------------------------------	---	-------------------------

Cała ramka CLIP FSK może wyglądać np. tak:

SMRR	MARK	MESSAGE						CRC	
		TYPE	LEN	Rekordy meldunku					
				TYPE	LEN	DATA	TYPE	LEN	DATA

Typowe rekordy:

- rekord czasu i daty (kod \$01)
- rekord numeru abonenta wywołującego (kod \$02)

Spotykane rekordy:

- rekord numeru wybranego przez abonenta wywołującego (kod \$03)
- rekord określający typ centrali (kod \$04)
- rekord opisu tekstowego (kod \$07)
- status centrali (kod \$13)

Poniżej przykładowe ramki CLIP. Wygubioną czcionką oznaczono pierwsze bajty ramek oraz sumę kontrolną.  
Przykładowy prosty meldunek może wyglądać tak (kody hex i ascii):

**HEX ASCII OPIS**

**\$80**                    **Wiadomość CLIP**  
**\$15**                    21 bajtów meldunku (łączenie z CRC)  
**\$01**                    **rekord daty i czasu**  
**\$08**                    8 bajtów w rekordzie  
**\$30**    '0'            znaki ascii układające się w datę 05-16 godz 17:35  
**\$35**    '5'            jeśli pierwszy bajt nr ma wartość \$50 (znak ascii 'P') oznacza to nr zastrzeżony  
**\$31**    '1'            jeśli pierwszy bajt ma wartość \$4F (znak ascii 'O') oznacza to, że abonent nie ma identyfikacji  
(np. jest podłączony do centrali analogowej)  
**\$36**    '6'  
**\$31**    '1'  
**\$37**    '7'

\$33	'3'	
\$35	'5'	
<b>\$02</b>		<b>rekord numeru abonenta wywołującego</b>
\$09		9 bajtów w rekordzie
\$30	'0'	znaki ascii nr abonenta wywołującego 012345678
\$31	'1'	
\$32	'2'	
\$33	'3'	
\$34	'4'	
\$35	'5'	
\$36	'6'	
\$37	'7'	
\$38	'8'	
<b>\$xx</b>		<b>CRC – suma modulo 2 wszystkich danych (wyłączając CRC)</b>

Niektóre centrale wysyłają dodatkowe ramki. Np. Siemens ESWD wysyła ramkę:

HEX	ASCII	OPIS
<b>\$80</b>		<b>Wiadomość CLIP</b>
\$31		liczba bajtów meldunku (łączenie z CRC)
<b>\$01</b>		<b>rekord daty i czasu</b>
\$08		8 bajtów w rekordzie
\$30	'0'	znaki ascii układające się w datę 05-16 godz 17:35
\$35	'5'	
\$31	'1'	
\$36	'6'	
\$31	'1'	
\$37	'7'	
\$33	'3'	
\$35	'5'	
<b>\$02</b>		<b>rekord numeru abonenta wywołującego</b>
\$09		9 bajtów w rekordzie
\$30	'0'	znaki ascii nr abonenta wywołującego '012345678'
\$31	'1'	
\$32	'2'	
\$33	'3'	
\$34	'4'	
\$35	'5'	
\$36	'6'	
\$37	'7'	
\$38	'8'	
<b>\$03</b>		<b>rekord nr wybranego</b>
\$07		7 bajtów w rekordzie
\$31	'1'	znaki ascii wybranego nr '1234567'
\$32	'2'	
\$33	'3'	
\$34	'4'	
\$35	'5'	
\$36	'6'	
\$37	'7'	
<b>\$04</b>		<b>rekord typu centrali</b>
\$11		17 bajtów w rekordzie
\$45	'E'	znaki ascii tekstu 'ESWD from SIEMENS'
\$53	'S'	
\$57	'W'	
\$43	'D'	
\$20	' '	
\$66	'f'	
\$72	'r'	
\$6F	'o'	
\$6D	'm'	

\$20 ' '  
\$53 'S'  
\$49 'I'  
\$45 'E'  
\$4D 'M'  
\$45 'E'  
\$4E 'N'  
\$53 'S'  
\$xx

#### **CRC – suma modulo 2 wszystkich danych (wyłączając CRC)**

Wielu może zdziwić po co wysyłać nr który wybrał abonent wywołujący? Przecież doskonale wiadomo do jakiej linii podłączony jest telefon. Zgadza się w 99% procentach przypadkach nr wybierany będzie taki sam jak nr linii telefonicznej, ale gdy abonent trafi do nas z przekierowanej rozmowy nr ten będzie inny. Przykładowo abonent o nr 1111111 przekierował połączenia na nr 2222222. Abonent 3333333 wybierając nr 1111111 dodzwoni się pod nr 2222222 a nie pod 1111111. W tej sytuacji na linie o numerze 2222222 zostanie wysłana następująca informacja:

Rekord \$01 - DATA i CZAS

Rekord \$02 – 3333333 (nr abonenta wywołującego)

Rekord \$03 – 2222222 (wybrany przez niego nr niezgodny z nr linii)

Rekord \$04 - ESWD from SIEMENS

Gdyby abonent 3333333 zadzwonił 1111111 bez włączonej funkcji przekierowania połączeń otrzymali byśmy poniższą ramkę:

Rekord \$01 - DATA i CZAS

Rekord \$02 – 3333333 (nr abonenta wywołującego)

Rekord \$03 – 1111111 (wybrany przez niego nr zgodny z nr linii)

Rekord \$04 - ESWD from SIEMENS

Podobnie będzie w przypadku posiadania numerów PBX. Dzięki rekordowi \$03 mamy namiastkę nr MSN/DDI z ISDN w ruchu przychodzącym na łączach analogowych. Dzięki rekordowi \$03 można odrzucać połączenia.

Algorytm odbioru CLIPa powinien być następujący:

1. Czekanie na SMRR
2. Odebranie MARK, nie MARK to skocz do 1
3. Zeruj CRC
4. Odebranie TYPE, jeśli różne od \$80 to skocz do 1
5. Odbierz LEN i zapamiętaj w zmiennej LenMsg
6. Odbierz TYPE, zmniejsz LenMsg
7. Odbierz LEN, zapamiętaj w zmiennej LenRec, zmniejsz LenMsg
8. Odbierz daną, zapisz w buforze, zmniejsz LenMsg i LenRec
9. Jeśli LenRec<>0 to skocz do 8
10. Jeśli znane pole TYPE to przygotuj do wyświetlenia/analizy (np. rekord daty/czasu można przygotować do zapisania w zegarze, jeśli pole nieznane nie interpretuj danych)
11. Jeśli LenMsg<>0 to skocz do 6
12. Odbierz CRC
13. Porównaj CRC, jeśli ok. to wyświetl dane

Taki algorytm zagwarantuje, że CLIP będzie wyświetlony bez względu na liczbę rekordów w komunikacie oraz ich kolejność. Na tej zasadzie działa CLIP w rejestratorze AVT-50xx i bez problemu radzi sobie z rekordami typu \$01, \$02 (data/czas/numer) jak i \$03, \$04, \$07, które przy wyświetlaniu informacji o nr pomija. Niestety nie można tego powiedzieć o niektórych konstrukcjach dekodatorów dostępnych w internecie, gdzie analiza kodu źródłowego dowodzi iż identyfikator zadziała tylko z ramkami typu \$01 i \$02. Pojawienie się dodatkowych ramek w komunikacie spowoduje odczytanie złej sumy kontrolnej w konsekwencji czego program nic nie wyświetli. Inna kolejność ramek może natomiast spowodować złe ustawienie zegara oraz wyświetlenie czasu zamiast numeru.

Jak wynika z powyższego tekstu urządzenie zbudowane i przetestowane u autora, który nie zagłębił się w odpowiednie normy może doprowadzić do sytuacji, że urządzenie będzie poprawnie działało tylko u niego!

Na koniec warto dodać, że obydwa standardy umożliwiają przekazanie numeru połączenia oczekującego (DIDCW – Calling Identity on Call Waiting). Po uaktywnieniu usługi, na czas przekazywania informacji o numerze rozmowa jest wyciszana. Operacje te przedstawiono na rysunku (**REDAKCJA: rysunek 10a, w pliku „cml612.pdf”**).

## Rozszerzone komendy AT

### Komendy modemów analogowych na przykładzie modemu MultiTech (DipModem):

Na początek musimy zapoznać się z komendami AT akceptowanymi przez modem. Nie jest to powtórzenie poprzednich rozdziałów ponieważ niektóre komendy działają w odmienny sposób oraz dodano wiele nowych ciekawych rozkazów. Warto zaznaczyć, że nie wszystkie popularne komendy modem akceptuje oraz to, że niektóre działają w trochę odmienny sposób. Wszystkie komendy poprzedzamy znakami AT. Komendę zatwierdzamy naciskając enter. Poniżej spis komend z komentarzami. Pogrubioną czcionką oznaczono standardowe ustawienia (po resece):

\$	wyświetla podstawowe ustawienia modemu
A/	powtórz ostatnią komendę
A	odebranie połączenia
Dnnnn	wybranie numeru nnnn. W komendzie poza cyframi można używać znaków specjalnych, oto ich znaczenie:
! lub &	wygenerowanie flasha, czas flasha ustawiamy w rejestrze S29 (standardowo 500ms)
, lub <	pauza 2 sekundy (ustawianie w rejestrze S8)
;	po wybraniu nr powraca do interpretacji komend, nie próbuje nawiązać połączenia z drugim modemem.
P	wybieranie impulsowe (akceptuje 0 1 2 3 4 5 6 7 8 9)
T	wybieranie tonowe (akceptuje * # A B C D 0 1 2 3 4 5 6 7 8 9)
W	czeka na ton wybierania przez czas ustawiony w S12 (standardowo 12 sekund)
En	echo komend 0 - wyłączone <b>1 - włączone</b>
H	rozłączenie połączenia (Hang-Up)
In	identyfikacja typu modemu 0 - rewizja 1 - wersja programu 6 - typ modemu
Mn	praca głośnika <b>0 - wyłączony</b> 1 - wyłączona po nawiązaniu połączenia 2 - głośnik zawsze włączony
Qn	kody zdarzeń <b>0 - włączone</b> 1 - wyłączone 2 - włączone podczas inicjalizacji połączenia, wyłączone gdy odbiera
Sn=x	zapis rejestru specjalnego "n" wartością "x"
Sn=?	wyświetla zawartość rejestru x (zawsze trzycyfrowo)
Vn	rodzaj kodów wynikowych 0 - numeryczna <b>1 - numeryczna i tekstowa</b>
Xn	rodzaj dzwonienia (wybierania) 0 - nie czeka na 400Hz, ignoruje zajętość, daje komunikat CONNECT gdy połączenie 1 - j/w + prędkość połączenia 2 - czeka na 400Hz, ignoruje zajętość, daje CONNECT gdy połączenie 3 - nie czeka na 400Hz, daje BUSY gdy zajętość oraz CONNECT gdy połączenie <b>4 - czeka na 400Hz, daje CONNECT gdy połączenie, BUSY gdy abonent zajęty lub NO DIAL</b> gdy brak sygnału z centrali przez 5 sekund

Yn	reakcja po odebraniu określonej liczby spacji <b>0 - brak reakcji</b> 1 - przerwanie połączenia
Z	sprzętowy reset
:Dn	monitor DTMF <b>0 – wyłączony</b> 1 – włączony
:Fn	Generator tonów <b>0 – wyłączony</b> 1 – włączony
:I	odczytuje źródło przerwania
:P	zapisuje pamięć ram modemu (upgrade), przykład: AT:Paaaa,xxxx,yyyy,... gdzie aaaa – adres w pamięci RAM, xxxx yyyy dane zapisywane pod kolejne adresy
:R	czyta U-rejestry, przykład: AT:Raa – odczytuje 16-bitowy rejestr z pod adresu aa
:U	zapisuje U-rejestry, przykład: AT:Uaa,xxxx,yyyy,zzzz,... – wpisuje do rejestrów użytkownika począwszy od adresu aa 16- bitowe dane xxxx yyyy zzzz itd.
+VCID=n	włączenie dekodera CLIP <b>0 – wyłączony</b> 1 – włączony tryb formatowany 2 – włączony format RAW
+VCDT=n	<b>0 - FSK po dzwonku</b> 1 – FSK zawsze oczekujący 2 – UK 3 – Japan 4 - DTMF
&\$	wyświetla ustawienia komend rozszerzonych AT&
&G	maksymalna prędkość transmisji 5 – 4.8kbps 6 – 7.2kbps 7 – 9.6kbps 8 – 12kbps <b>9 – 14.4kbps (standardowe dla modemu SI2414)</b> 10 – 16.8kbps 11 – 19.2kbps 12 – 21.6kbps 13 – 24kbps 14 – 26.4kbps 15 – 28.8kbps 16 – 31.2kbps <b>17 – 33.6kbps (standardowe dla modemu SI2433)</b>
&H	wybór standardu połączenia <b>0 – V.90 z automatyczną zmianą protokołu (56kbps do 300bps) (standard dla SI2456)</b> 1 – tylko V.90 (56kbps do 28kbps) <b>2 – V.34 z automatyczną zmianą protokołu (33.6kbps do 300bps) (standard dla SI2433)</b> 3 – tylko V.34 (33.6kbps do 2400bps) <b>4 – V.32bis z automatyczną zmianą protokołu (14.4kbps do 300bps) (standard dla</b> <b>SI2414)</b> 5 – tylko V.32bis (14.4kbps do 480bps) 6 – tylko V.22bis (2400bps lub 1200bps) 7 – tylko V.22 (1200bps) 8 – tylko Bell 212 (1200bps) 9 – tylko Bell 103 (300bps) 10 – tylko V.21 (300bps) 11 – tylko V.23 (1200/75bps)
&Tn	testowanie modemu 0 – terminate test mode 2 – test DSP i DAA (zapętla sygnał analogowy)

	3 – j/w dodatkowo testuje układ SI3015)
	6 – test sumy kontrolnej
&Z	przejdzie w tryb obniżonego poboru mocy
%%\$	wyświetla ustawienia komend rozszerzonych AT%
%Cn	kompresja
	0 - wyłączona
	<b>1 - włączona MNP</b>
	2 - włączona V.42
	3 - V.42 i MNP
%On	tryb odbioru
\\$	wyświetla ustawienia komend rozszerzonych AT\
\Bn	liczba bitów znaku
	0 – 6N1
	1 – 7N1
	2 – 7P1
	<b>3 – 8N1</b>
	5 – 8P1
	6 – 8X1
\Nn	wybór protokołu
	0 – Wire mode (bez korekcji błędów, bez kompresji)
	<b>3 – V.42 automatyczny wybór protokołu</b>
\Pn	typ parzystości
	0 – even
	1 – space
	2 – odd
	3 – mark
\Qn	sposób kontroli przepływu
	0 – wyłączona kontrola
	<b>2 – tylko sygnał CTS</b>
	3 – para sygnałów RTS/CTS
	4 – potwierdzenia Xon/Xoff
\Tn	szybkość transmisji do terminala
	0 – 300bps
	1 – 600bps
	2 – 1200bps
	3 – 2400bps
	4 – 4800bps
	5 – 7200bps
	6 – 9600bps
	7 – 12.0kbps
	8 – 14.4kbps
	<b>9 – 19.2kbps</b>
	10 – 38.4kbps
	11 – 57.6kbps
	12 – 115.2kbps
	13 – 230.4kbps
	14 – 245.760kbps
	15 – 307.200kbps
\U	gdy modem pracuje w trybie szeregowym (UART'owym) wysyła impuls 25ms na wyjściu RI i
DCD. INT	przyjmie zanegowany poziom ESC, RTS zanegowany CTS.
	Gdy modem pracuje w trybie równoległym, wyśle impuls 25ms na wyjściu INT.
\Vn	typ komunikatów
	0 – raporty o połączeniu i protokole
	<b>1 – raporty tylko o połączeniu</b>

## S-Rejestry

Modem posiada oczywiście S-rejestry.

Rejestr 0 odpowiada za to, po ilu dzwonekch modem automatycznie odbierze połączenie. Jeśli wpisze 0 modem nie odbierze połączenia. Ustawienia tego rejestru mogą być mylące, ponieważ połączenie może odebrać program terminala komendą "ATA" po ustawionej w nim liczbie dzwonekch. Należy uważać z



ustawieniem różnym od zera, ponieważ modem nie wysyła wtedy komunikatu RING tylko CONNECT lub NO CONNECT. Standardowe ustawienie 0.

Rejestr1	licznik odebranych dzwońków
Rejestr2	kod znaku escape, standardowo 43 (znak "+")
Rejestr3	kod znaku CR, standardowo 13
Rejestr4	kod znaku LF, standardowo 10
Rejestr5	kod znaku bakspace, standardowo 8
Rejestr6	czas oczekiwania na sygnał zgłoszenia centrali w sekundach, standardowo 3
Rejestr7	czas na nawiązanie połączenia. Jeśli w zadanym czasie modem nie nawiąże połączenia, rozłączy się. Standardowe ustawienie 60
Rejestr8	czas trwania pauzy (znak ", " lub „,<”) w sekundach, standardowo 2
Rejestr9	(std=6) 1/10 sek.
Rejestr10	opóźnienie od zmiany stanu linii DCD na nieaktywny do rozłączenia (zwolnienia linii telefonicznej) w 1/10 sek, standardowo 6 (0.6sek)
Rejestr12	czas od wysłania kodu ESC (standardowo "+++") do otrzymania komunikatu "OK" w 1/50 sek. standardowo 50

## U-Rejestry

Modem poza S-rejestrami posiada jeszcze specyficzne dla niego 16-bitowe U-Rejestry. Rejestrów tych jest dość dużo dlatego omówię najważniejsze z nich:

Pierwsze U-Rejestry ustawiają filtry dekodera DTMF do wymagań w danym kraju. W PDF'ie znajduje się spis ustawień dla wielu krajów. Lista ta jest dość długa i co cieszy, znajduje się na niej Polska.

W rejestrach U37-U40 ustawiamy liczbę impulsów odpowiadającą cyfrze wybieranej w systemie impulsowym. U42-U45 ustala czas przerwy/zwarcia i czas pomiędzy wybieranymi cyframi w systemie impulsowym.

U47-U48 ustalają czas trwania i czas przerwy tonów DTMF.

Bit 1 w rejestrze U67 decyduje o minimalnym napięciu dzwonienia. Wartość 0 rejestru wybiera napięcie 11-22V, wartość 1 – 17-33V. Jeśli modem zmienia stan linii RI podczas występowania impulsów zakłócających na linii warto bit ten ustawić na 1.

Znaczenie pozostałych rejestrów jest opisane w PDF'ie.

## Wykrywanie podłączonego równoległe telefonu

Wydając komendę AT:U69,0004 włączamy monitorowanie napięcia na linii telefonicznej. Napięcie to możemy odczytać komendą AT:R79. Podczas testów przy wolnej linii telefonicznej odczytałem wartość 0018. Jeśli słuchawka przyłączona równoległe telefonu była podłączona, odczytałem wartość 0004. Jeśli linia była odłączona (uszkodzona) odczytywałem wartość 0000. Jeden bit odpowiada napięciu 2,75V. Jak łatwo się zorientować na testowanej linii przy odłożonej słuchawce panowało napięcie 77V, przy podniesionej 11V. Funkcja monitorowania napięcia może być wykorzystana także do wykrycia faktu uszkodzenia linii telefonicznej.

## Upgrade

Do upgrade oprogramowania służy komenda AT:Paaaa,xxxx,yyyy,... gdzie aaaa – adres w pamięci RAM, xxxx yyyy dane zapisywane pod kolejne adresy. Jak więc widać od producenta ściągamy najnowszą wersję programu w postaci pliku tekstowego, wysyłamy do modemu, po czym wykonujemy komendę ATZ. Jeśli sumy kontrolne będą się zgadzały, modem wykona program z pamięci RAM, jeśli nie z ROM.

## **Komendy modemów ISDN na przykładzie TA128**

Opisywany modem ISDN nie ma zbyt wielu komend AT, ale są one odmienne od standardowych komend stosowanych w modemach analogowych. Zmiany dotyczą głównie programowania nr MSN, ustawiania protokołów transmisji oraz reakcji modemu na wybrany protokół transmisji danych.

**ATA** Odebranie przychodzącego wywołania.

**ATBx** Ustawienie protokołu kanału B.  
X=0 64K HDLC (domyślny).  
X=3 X.75 Transparent, podobnie jak ATB30.  
X=4 Konwersja Async PPP to Sync PPP

X=5	Bit Transpacy (Ta komenda jest używana jedynie przy wykorzystaniu soft-G3Fax w programie RVS-COM).
X=11	V.110 Async. Prędkość=1200 bps (opcja).
X=12	V.110 Async. Prędkość=2400 bps (opcja).
X=13	V.110 Async. Prędkość=4800 bps (opcja).
X=14	V.110 Async. Prędkość=9600 bps (opcja).
X=15	V.110 Async. Prędkość=19200 bps (opcja).
X=16	V.110 Async. Prędkość=38400 bps (opcja).
X=20	V.120 Async.
X=30	X.75 Transparent.
X=31	X.75 T.70 NL
X=32	X.75 ISO 8208
X=41	Konwersja Async PPP to Sync PPP w trybie ML-PPP (kompatybilne z Microsoft ISDN Accelerator Pack)
X=42	Konwersja Async PPP to Sync PPP w trybie ML-PPP z funkcją "pasmo na żądanie".
<b>ATCODEC</b>	Ustawienie lub wyświetlenie rodzaju kodeka w porcie analogowym.
ATCODEC=n	n=0 dla A-Law n=1 dla U-Law
ATCODEC?	Wyświetlenie bieżących ustawień.
<b>ATD</b>	Wybranie numeru.
ATD0202422	Wybranie numeru 0202422.
<b>ATDL</b>	Powtórzenie wybierania ostatniego numeru.
<b>ATEx</b>	Wyświetlanie wprowadzonych znaków na ekranie w trybie AT (tzw. Echo)
X=0	Echo wyłączone
X=1	Echo włączone
<b>ATHx</b>	Załączenie/ Wyłączenie sygnału linii.
X=0	Wyłączenie (to samo co ATH) - powoduje również zerwanie połączenia.
X=1	Załączenie.
<b>ATIx</b>	Wyświetlenie informacji o adapterze.
X=0	Wyświetlenie informacji o wersji adaptera, bieżącym protokole i prędkości połączenia.
X=1	Wyświetlenie następujących informacji: typ centrali, kodek, SPID,MSN.
X=2	Wyświetlenie informacji o ostatnio nawiązanych połączeniach głosowych (poprzez interfejsy POTS).
X=3	Wyświetlenie informacji o ostatnio nawiązanych połączeniach (transmisja danych).
<b>ATO</b>	Przełączenie z trybu komend do trybu on-line
<b>ATP</b>	Ustawienie lub wyświetlenie kraju, lub typu centrali.
<b>ATP=n</b>	
N=1	Belgia
N=2	Chiny
N=3	Kolumbia
N=4	Dania
N=5	Holandia
N=6	Euro ISDN
N=7	Finlandia

N=8	Francja
N=9	Niemcy
N=10	Izrael
N=11	Włochy
N=12	Japonia
N=13	Korea
N=14	Korea
N=15	Singapur
N=16	Słowenia
N=17	Południowa Afryka
N=18	Hiszpania
N=19	Szwecja
N=20	Szwajcaria
N=21	Tajwan
N=22	Wielka Brytania
N=23	USA(AT&T Multi-P)
N=24	USA(AT&T P-T-P)
N=25	USA(AT&T NI-1)
N=26	USA(AT&T NI-2)
N=27	USA(AT&T NT1/DMS)
ATP=?	Wyświetlanie aktualnie wybranego kraju lub typu centrali.
<b>ATQx</b>	Zwracanie kodów wynikowych.
X=0	Zwracaj kody wynikowe.
X=1	Nie zwracaj kodów wynikowych.
<b>ATS</b>	Ustawienie lub wyświetlenie wartości rejestru.
ATS0=1	Wpisanie do rejestru S1 wartości "0" (domyślnie S0=0-brak automatycznej odpowiedzi na wywołania).
ATSr?	Wyświetlanie aktualnej wartości rejestru r.
ATS1?	Rejestr S1 nie może być zapisywany, zawiera licznik dzwoneń.
ATS2	Kod znaku "Escape" (domyślnie S2=43,ASCII "+").
ATS3	Kod znaku "carriage return" (powrót karetki)(domyślnie S3=13).
ATS4	Kod znaku "line feed"(przejdź do następnej linii) (domyślnie S4=10,"CTRLJ").
ATS5	Kod znaku "back space"(domyślnie S5=8,"CTRLH").
ATS7	Czas oczekiwania na nośną po wybraniu numeru (domyślnie S7=30 sekund).
ATS12	Czas oczekiwania na kod "escape" (domyślnie S12=50 =5 sekund).
ATS25	Czas rozpoznawania sygnału DTR (domyślnie S25=5 =0,5 sekundy).
ATS30	Czas po którym następuje automatyczne zerwanie połączenia, jeśli nie występuje transmisja danych n*10 sekund (n=0 do 255, domyślnie s30=0 co nie powoduje zrywania połączenia).
ATS37	Wysyłanie informacji "Low layer Compatability" S37=0 - nie wysyłaj LLC (domyślne), S37=1 - wysyłaj LLC.
ATS38	Rozmiar okna dla HDLC 56K lub 64K (domyślnie 7).
ATS39	Rozmiar pakietu dla HDLC 56K lub 64K, od 1 do 2048 (domyślnie 1024).
ATS40	Rozmiar okna dla V.120 (domyślnie 7).
ATS41	Rozmiar pakietu dla V120 (domyślnie 256).
ATS44	Rozmiar okna dla X.75 (Transparent) (domyślnie 2).
ATS45	Rozmiar pakietu X.75 (Transparent),od 1 do 2048 (domyślnie 2).
ATS46	Rozmiar okna dla X.75 T.70 NL (domyślnie 2).
ATS47	Rozmiar pakietu X.75 T.70 NL,od1 do 2048 (domyślnie 128).
ATS50	Rozmiar okna dla X.75 ISO 8208 (domyślnie 2).
ATS51	Rozmiar pakietu X.75 ISO 8208, od 128 do 2048 (domyślnie 1024).
ATS53	Od 1000 do 7000 bajtów (domyślnie 4 co oznacza 4000 bajtów). Zestawienie połączenia w drugim kanale B przy połączeniu za pomocą protokołu ML-PPP BOD (ATB42) następuje, jeśli zapisana

		w tym rejestrze średnia prędkość transmisji utrzymuje się przez min. 10 sekund.
<b>ATS54</b>		Okres czasu, od 5 do 20 sekund (domyślnie 5), po którym następuje zwolnienie drugiego kanału B (zerwanie połączenia w tym kanale) jeśli prędkość transmisji znajdzie się poniżej wartości zapisanej w rejestrze S53, (tylko w połączeniach przy protokole ML-PPP BOD).
<b>ATS55=n</b>		Wybór sposobu dzwonienia dla urządzenia podłączonego do interfejsu POTS 1,(n=0 do 7)
	"n"	Ring On Ring Off
	0	0,5 sekundy 0,5 sekundy
	1	0,5 1,0
	2	0,5 1,5
	3	1,0 1,0
	4	1,0 2,0 (domyślne)
	5	1,0 3,0
	6	2,0 2,0
	7	2,0 3,0
<b>ATS56=n</b>		Wybór sposobu dzwonienia dla urządzenia podłączonego do interfejsu POTS 2, (n=0 do 7), tabela jak dla POTS 1.
<b>ATSTBY=n</b>		Ustawia czas przejścia w tryb "standby", n=3 do 10 (domyślnie 3).
<b>ATSBY?</b>		Wyświetla wartość "n".
<b>ATUPG</b>		Uaktualnienie oprogramowania firmowego adaptera, należy postępować zgodnie ze wskazówkami ukazującymi się na ekranie.
<b>AVTx</b>		Wybór postaci wyświetlanych kodów wynikowych.
	X=0	Wyświetlaj kody w postaci cyfr.
	X=1	Wyświetlaj kody w postaci słów.
<b>ATXn</b>		Załącza / wyłącza zwracanie rozszerzonych kodów wynikowych.
	N=0	Wyłącz j.w.
	N=1	Załącz j.w.
<b>ATZn</b>		Resetuje adapter i powoduje powrót do ustawień zawartych w profilu 0.
<b>ATZn</b>		Resetuje adapter i powoduje powrót do ustawień zawartych w profilu n, n=0 lub 1.
<b>AT&amp;ABx</b>		Wykrywanie prędkości.
	X=0	Automatycznie wykrywaj prędkość portu szeregowego, (domyślne).
	X=1	Nie wykrywaj prędkości, używaj aktualnych ustawień portu.
<b>AT&amp;AB?</b>		Wyświetl aktualne parametry portu szeregowego.
<b>AT&amp;APx</b>		Wykrywanie protokołu nadchodzącego wywołania.
	X=0	Wyłącz automatyczne wykrywanie.
	X=1	Włącz automatyczne wykrywanie.
<b>+++</b>		Komenda powoduje wygenerowanie kodu "escape".
<b>AT&amp;Cx</b>		Kontrola sygnału DCD.
	X=0	Dygnął DCD, domyślnie zawsze aktywny, (domyślne).
	X=1	Sygnał DCD aktywny po nawiązaniu połączenia.
<b>AT&amp;Dx</b>		Kontrola sygnału DTR.
	X=0	Ignoruj sygnał DTR, domyślnie zawsze aktywny.
	X=2	Zaniknięcie sygnału DTR powoduje zerwanie połączenia,(domyślne).
<b>AT&amp;Ex</b>		Wybór prędkości transmisji w kanale B.
	X=0	64K bps (domyślne).
	X=1	56K bps.

<b>AT&amp;F</b>	Resetowanie rejestrów do ustawień fabrycznych.
<b>AT&amp;Kx</b>	Kontrola przepływu DTE/Modem.
X=0	Wyłącz kontrolę przepływu DTE//DCE.
X=3	Włącz kontrolę przepływu DTE/DCE - RTS/CTS,(domyślne).
X=4	Włącz kontrolę przepływu DTE/DCE - XON/XOFF.
X=6	Włącz kontrolę przepływu DTE/DCE - RTS/CTS oraz XON/XOFF.
<b>AT&amp;Nx</b>	Wybierz tryb transmisji voice.
X=0	Wybierz "mowa".
X=1	Wybierz 3.1K audio.
<b>AT&amp;Tn</b>	Ta komenda jest używana jedynie w procesie homologacji.
<b>AT&amp;TEST</b>	Uruchomienie auto-testu.
<b>AT&amp;V</b>	Wyświetlenie zawartości profili Y1 i Y2.
<b>AT&amp;Wn</b>	Zapisz aktualną konfigurację profilu n.
<b>AT&amp;Yn</b>	Załaduj konfigurację zawartą w profilu n.
<b>AT&amp;Zir=n*m*p</b>	Ustawienie parametrów decydujących o reakcji adaptera na przychodzące wywołania."n" - lokalny numer telefonu (MSN), "*"- symbol pod-adresu, "m"- pod adres. Druga "*" - jest opcjonalna, "p"- identyfikator protokołu. Jeśli p=0 są akceptowane wszystkie dostępne protokoły (domyślne). p=1 odbiór tylko V.110 (opcja) =2 odbiór tylko V.120 =3 odbiór tylko V.75 Transparent =4 odbiór tylko X.75 T.70NL =5 odbiór tylko X.75 EFT =6 odbiór tylko HDLC(PPP...). r=1 do 5, 0 dla POTS 1, 1 dla POTS 2, 2 do 5 dla danych. Maksymalna długość numeru MSN wynosi 18, dla pod-adresu 8. Na przykład jeśli chcesz, aby były przyjmowane wywołania z protokołem X.75 Transparent komenda może wyglądać tak AT&ZI2=81722043**3(bez pod-adresu).
<b>AT&amp;ZI?</b>	Wyświetla bieżące ustawienia przychodzących wywołań.
<b>AT&amp;Zor=n*m</b>	Ustawia numery MSN i pod-adresy dla portów POTS 1, POTS 2 oraz portu danych dla wywołań wychodzących. "n"- lokalny numer telefonu (MSN), "*" - symbol pod-adresu, "m"- pod-adres.
<b>AT%Sn</b>	Parametry portu szeregowego. N=0,1.
N=0	Tryb automatycznego wyboru prędkości (domyślny) - 300, 1200,2400,4800,9600,19200,38400,57600 lub 115200 bps.
N=1	Ustawienie stałe DTE=230400 bps.
<b>AT#Cn</b>	Ustawienie identyfikacji wywołującego (CallerID).
N=0	Wyłącz „CallerID” (domyślnie)
N=1	Włącz „CallerID”
N=2	Pozwól na wyświetlanie CallerID, CallerSub, CLIP, CalledID, CalledSub
<b>AT#C?</b>	Wyświetl aktualne ustawienia.
<b>A/</b>	Powtórz ostatnią komendę
<b>AT@</b>	Uruchom program konfiguracyjny adaptera.

## Upgrade modemu TA128

Pierwszym krokiem jest ściągnięcie najnowszej wersji oprogramowania ze strony producenta. Jest to plik z rozszerzeniem „.ABS”. W Internecie można znaleźć programy które przeprowadzają upgrade. Mają one jednak poważną wadę, spowodowaną tym, że przed uaktualnieniem sprawdzają wersję programu. Gdy programu w modemie nie ma nie potrafią przeprowadzić upgrade. Sytuacja taka może mieć miejsce gdy operacja upgrade zostanie przerwana. Wtedy to modem czeka na nową wersję programu i nie reaguje na żadne inne komendy! Stan taki jest sygnalizowany miganiem diod AA. Gdy taka sytuacja przydarzy się nam lub po prostu chcemy wiedzieć jak wygląda upgrade musimy zrobić to z programu terminala. Od razu zaznaczam, że nie uda się to z Terma90. Upgrade sprawdziłem natomiast w HyperTermie dla PC i Termie4.6 dla Amigi i powiodło się bez problemów. Podczas upgrade znaczenie mają parametry transmisji, które należy ustawić następująco:

Baud Rate (prędkość transmisji) 115200  
Data Bit (bitów danych) 8  
Parity (parzystość) NONE (Brak)  
Stop Bit (bitów stopu) 1  
Flow Control (sterowanie przepływem) RTS/CTS (Sprzęt)

Wydajemy komendę „AT@”. Naszym oczom ukaże się menu:



### (PLIK „TAE1.JPG”)

Naciskamy klawisz **1**. Modem odpowiada:

Are you sure to upgrade the new firmware <Y/N>:\_

Gdy naciśniemy **N** wrócimy do menu. Gdy naciśniemy **Y** na ekranie ujrzymy:

#### **ISDN TA upgrade procedure starts:**

Erase current driver <Y/N>:\_

To ostatnia chwila aby wycofać się z upgrade. Gdy naciśniemy **Y** nastąpi skasowanie programu w pamięci flash.

Podczas kasowania na ekranie zobaczymy:

**Erasing ISDN driver..... (wait about 5 sekonds)**

#### **Finishing erase procedure**

#### **Waiting for new firmware through ASCII mode transmission**

**If you want to exit upgrade, press '\$' to exit now.**

Zaznaczam od razu, że po skasowaniu programu wyjście przez naciśnięcie klawisza **\$** nie ożywi modemu.

Jedyna czynność jaka będzie on potrafił wykonać to skasować flash i zapisać nową zawartością. Mylący trochę jest też komunikat „**Waiting for new firmware through ASCII mode transmission**”, który sugeruje, że plik należy wysłać wybierając w menu terminala „Transmisja/Wyślij plik tekstowy”. Otóż nie! Plik należy wysłać wklejając go w okno terminala. Aby to zrobić należy plik z rozszerzeniem ABS wczytać do edytora ascii (np. TextPad dla PC czy CED dla Amigi). Następnie należy zaznaczyć cały tekst i skopiować do schowka. Tekst taki wklejamy do okna terminala. Tu kolejna niespodzianka od Microsoftu. Po naciśnięciu kombinacji CTRL+V tekst nie zostanie wklejony choć wynikałoby to ze skrotu klawiszowego (na Terminalu dla Amigi nie ma takiego problemu). Na szczęście wklejenie można wykonać wybierając w menu „Edycja/Wklej do hosta”. Po tej operacji nastąpi upgrade. Trwa ono około 20 minut. Podczas upgrade w oknie terma widzimy:

#### **Compare S0-rekord OK.**

**Load Addr = #####**

gdzie za tekstem „Load Addr =” ukazuje się adres zapisywanego rekordu. Osoby zorientowane w programowaniu różnych procesorów na podstawie tekstu „S-rekord” mogą podejrzewać, że w modemie siedzi jakaś Motorola. I nie mylą się. Modem napędza Motorola 68302 z zegarem 14MHz. PC’owcy zadadzą pytanie „Dlaczego nie Intel?”. Odpowiedź jest prosta, w modemie wymagana jest szybkość i pewność, a tego Intel w prosty i tani sposób nie może zapewnić. Jak to duża moc obliczeniowa i 14MHz? O mocy Motorolli może świadczyć fakt, że procesor MC68302/25MHz potrafi w czasie rzeczywistym dekodować programowo dwa strumienie MP3! Jaki byłby potrzebny do tego Pentium?

**Wracając do tematu, po udanym upgrade ujrzymy komunikat:**

#### **Finish upgrade procedure**

**Press 'N' to exit this program and go into AT command mode.**

### **ISDN TA upgrade procedure starts:**

Erase current driver <Y/N>:\_

Oczywiście naciskamy **N** i mamy zainstalowane nowe oprogramowanie.

Na koniec uwaga dotycząca sytuacji gdy w modemie nie ma oprogramowania. Wtedy to po włączeniu wchodzi on do upgrade wysyłając tekst:

### **ISDN TA upgrade procedure starts:**

Erase current driver <Y/N>:\_

Oczywiście jest, że tego tekstu nie zobaczymy ponieważ program terminala będzie uruchomiony na długo po włączeniu zasilania modemu. Jeśli jest to modem zewnętrzny to wystarczy na chwilę wyłączyć mu zasilanie aby ujrzeć tekst, albo „w ciemno” nacisnąć **Y**.

**UWAGA!**

Nie zalecam programów do upgrade napisanych pod Win. Nie są one zbyt inteligentne i w przypadku przerwania procedury upgrade nie będą potrafiły wgrać oprogramowania. Upgrade przez terminal jest proste i pewne. Poza tym skuteczne na wszystkich platformach sprzętowych i programowych.

## ***Komendy modemów SDI***

Modem SDI interpretuje kilka komend AT ale odpowiada na wszystkie standardowe komendy, choć ich nie wykonuje. Właściwie najistotniejszą komendą w SDI jest ATD. Jakkolwiek nr podany za tą komendą i sposób wybierania nie jest istotny to jest ona potrzebna aby modem „połączył się” z Internetem. Modem SDI po prostu symuluje modem analogowy czy ISDN, ale służy tylko i wyłącznie do połączeń z Internetem. Po wydaniu komendy **ATD** łączy się z dostawcą usług Internetowych, następnie kontrolę na transferem danych pełni stos TCP/IP. Taka realizacja połączenia z Internetem ma tą zaletę, że nie ma problemów z podłączeniem modemu do różnych platform sprzętowych. Wymagany jest jedynie port RS232C, standardowy sterownik modemu oraz program obsługujący stos TCP/IP. Krótko mówiąc przy zmianie z modemu (analogowego czy ISDN) na SDI nie są wymagane (prawie) żadne zmiany.

Komenda **ATH** przerywa połączenie po uprzednim wydaniu komendy ESCAPE (+++). Podczas podłączania modemu SDI należy zwrócić uwagę aby przed modemem nie były włączone żadne urządzenia. Co prawda gdy wyłączymy zasilanie modemu SDI linia z cyfrowej zamienia się w analogową i można na niej przeprowadzić połączenia telefoniczne zwykłym telefonem, to jednak w przypadku gdy linia pracuje jako cyfrowa włączone w nią telefony analogowe będą przeszkadzać w transmisji danych i głosu.

## ***Komendy modemów GSM na przykładzie modemu Wismo firmy WaveCom:***

### **Fotografia (strona 3 pliku „WaveCom Integra.pdf”)**

W tym rozdziale zajmujemy się modułami GSM. Wybrałem moduł firmy WaveCom. Są to dobre moduły a ponadto osiągalne w promocyjnych cenach u niektórych operatorów. Dużą zaletą modułów jest to, że można wykorzystać jego pamięć i procesor do własnych celów. Dzięki temu w prostych zastosowaniach moduł bez pomocy zewnętrznego procesora może realizować różne zadania (np. pomiar temperatury i przekazanie wyników SMS'em). Ciekawostką jest fakt, że za drobną opłatą (około 5 euro) w modemie zostaną zaimplementowane komendy AT obsługujące stos TCP/IP. Dzięki nim korzystanie z GPRS'a jest dziecinnie łatwe. Na początek proponuje moduł FastRack. Jest to kompletny modem wymagający tylko anteny i zasilania. Modem wyposażony jest w gniazdo karty SIM, antenowe, zasilania i port RS232C. Nie ma więc problemu aby podłączyć modem do komputera z RS232C. O tym, że może to być dowolny komputer świadczy fakt, że większość testów przeprowadzono Amigą.

W seryjnej produkcji korzystniejsze jest zastosowanie modułu QUIK. Jest to bardzo mały moduł, zasilany napięciem 3,6V, stosowany także w telefonach komórkowych.

## **Inicjalizacja modułu**

Przed rozpoczęciem pracy z modułem należy go zainicjalizować (np. wpisać pin). Opisze tu obsługę modemu z poziomu programu terminala. Gdy będziemy chcieli podłączyć modem do mikrokontrolera należy pamiętać, że sekwencje obsługi linii TXD, RXD, RTS, CTS są takie same jak w modemach analogowych. Modem dostarczony przez producenta ma ustawione następujące parametry transmisji:

**Prędkość: 9600**

**Format ramki: 8 N 1**

i takie parametry należy ustawić w programie terminala. Aby sprawdzić czy dobrze ustawiliśmy parametry transmisji wydajemy komendę:

**AT**

Modem powinien odpowiedzieć:

**OK.**

Jeśli tak nie jest należy przeczytać dalszą część rozdziału poświęconą ustawieniu parametrów transmisji. Gdy parametry transmisji są dobrze ustawione możemy przystąpić do inicjalizacji, która wygląda następująco: wydajemy komendę:

**AT+CPIN?**

W odpowiedzi może my ujrzeć jeden z komunikatów:

**+CPIN: SIM PIN**           oznaczający, że wymagane jest wpisanie kodu PIN  
**+CPIN: SIM PUK**           oznaczający, że wymagane jest wpisanie kodu PUK  
**+CPIN: OK**                oznaczający, że nie jest wymagane wpisanie kodu

Aby wpisać kod PIN należy wydać komendę:

**AT+CPIN=xxxx** gdzie xxxx właściwy kod

Ujrzymy odpowiedź:

**ERROR**                   jeśli wpisaliśmy zły kod PIN, lub

**OK**                        gdy kod będzie poprawny.

Jeśli konieczne jest wpisanie kodu PUK spowodowane trzykrotnym wpisaniem niepoprawnego kodu PIN wydajemy komendę:

**AT+CPIN=xxxxxxxx,yyyy**       gdzie xxxxxxxx kod PUK, yyyy nowy kod pin

Po wpisaniu PIN'u modem zaloguje się do sieci. Objawia się to miganiem diody w modemie oraz komunikatem **+WIND: 4**

Rozkazem **AT+CSQ** możemy zmierzyć poziom sygnału docierający do anteny. W odpowiedzi otrzymamy dwie liczby. Jeśli są to **99,99** oznacza, że nie można zmierzyć sygnału (jesteśmy poza zasięgiem). Wartość pierwszej liczby mniejsza niż 15 oznacza słaby sygnał i mogą być problemy z łącznością.

Komendą **AT+CMEE=1**, **AT+WIND=63** i **AT+CRC=1** włączamy szczegółowe komunikaty o błędach i szczegółowe informacje o procesie połączenia. Zwłaszcza ta druga komenda jest przydatna przy połączeniach audio, gdzie łatwo możemy określić w jakim stanie (łączenie, wysyłanie sygnału dzwonka, itp) znajduje się modem. Należało by jeszcze ustalić w jakim formacie chcemy otrzymywać i wysyłać SMS'y. Służy do tego komenda **AT+CMFG=1** konfiguruje SMS jako tekst. SMS'y można także skonfigurować w formacie RAW, ale jest to format mniej przejrzysty dla człowieka.

Komenda **AT+CNMI=2,1,0,0,0** powoduje, że przychodzące SMS'y będą zapisywane na karcie SIM, a my powiadamiani o fakcie ich odebrania. Po przeczytaniu SMS'a z pamięci będziemy mogli go skasować lub pozostawić. Komenda **AT+CNMI=2,2,0,0,0** spowoduje, że przychodzące SMS'y nie będą zapamiętywane na karcie SIM tylko wysyłane na port RS.

Zapomniałem jeszcze wspomnieć o komendzie **AT+CLIP=1** włączającą identyfikację połączeń przychodzących. Dzięki temu można odrzucać połączenia z niechcianym numerów przez co nasze zdalnie obsługiwane urządzenie będzie bardziej odporne na próby „włamania”.

Prędkość transmisji ustawiamy komendą **AT+IPR**. Aktualne ustawienie można odczytać przez:

**AT+IPR?**

Możliwe do ustawienia prędkości możemy poznać wydając komendę:

**AT+IPR=?**

Zalecam ustawienie **AT+IPR=0** uaktywniające automatyczne wykrywanie prędkości transmisji. Komendą **AT+ICF** ustawiamy format ramki. Podobnie jak dla komendy **AT+IPR** i tu możemy podejrzeć aktualne ustawienia, jak i możliwe do wpisania wartości.

Konfigurację dotyczącą np. SMS'ów można zapisać w pamięci modemu komendą **AT+SAS**. Część ustawień (aby zobaczyć jakich należy wydać komendę **AT&V**) zapisujemy komendą **AT&W**. Należy jednak pamiętać, że nie wszystkie ustawienia zostaną zapisane (np. CLIP).

Dla ciekawskich proponuje wydanie sekwencji rozkazów **AT+COPS=3,0** i **AT+COPS?** w wyniku, której otrzymamy informacje o naszym operatorze.

## Transmisja danych

Transmisja danych przebiega dokładnie tak samo jak i w przypadku modemów analogowych. Po wydaniu komendy **ATDxxxxx** nastąpi próba połączenia z drugim modemem. Warto wspomnieć, że modem, z którym łączymy się nie musi być modemem GSM. Może to być modem analogowy lub cyfrowy. W trakcie nawiązywania połączenia ujrzymy komunikaty:

**+WIND: 5,1**               oznaczający rozpoczęcie procesu zestawiania połączenia



**+WIND: 2**                    oznaczający wysyłanie sygnału dzwonienia  
**BUSY**                        oznaczający, że abonent żądany jest zajęty

Wybieranie możemy przerwać wysyłając komendę **ATH**.

Gdy dojdzie do połączenia ujrzymy komunikat **CONNECT 9600**. Połączenie rozłączamy tak samo jak w przypadku modemu analogowego wysyłając **+++** po otrzymaniu **OK ATH**.

Podczas połączenia przychodzącego otrzymamy komunikat

**RING** lub jeśli włączyliśmy szczegółowe raporty komendą **AT+CREG=1** komunikat:

**+CRING: xxxx**, gdzie xxxx typ połączenia (np. VOICE, DATA, FAX). Jeśli włączyliśmy **CLIP** pojawi się dodatkowo komunikat:

**+CLIP" „xxxxxxx”,145,,,”yyyy”**

gdzie xxxxxx nr abonenta wywołującego, yyyy wpis w książce przypisany do danego numeru. Połączenie odbieramy rozkazem **ATA**. Jeśli nie chcemy przyjąć połączenia wydajemy rozkaz **ATH**. W modemie GSM tak jak w przypadku innych modemów istnieje rejestr **S0** w którym możemy ustawić po ilu dzwonekch modem ma odebrać połączenie.

## Wysyłanie i odbieranie SMS'ów

O konfiguracji modemu dla SMS'ów wspominałem już w poprzednich rozdziałach. Zakładam, że modem skonfigurowano dla SMS tekstowych zapamiętywanych na karcie SIM. Aby wysłać SMS należy wydać rozkaz: **AT+CMGS="xxxx"**                    gdzie xxxx nr na który chcemy wysłać SMS'a (dla testu możemy wysłać SMS'a do samego siebie)

Modem odpowie znakiem zachęty:

>

wpisujemy po nim treść SMS'a. Naciskając kombinację **CTRL+Z** (znak o kodzie \$1A) powodujemy wysłanie SMS'a. Jeśli chcemy zrezygnować z wysyłania naciskamy klawisz **ESC** (kod \$1B). Po chwili dostaniemy odpowiedź:

**+CMSS: xxx**                    gdzie xxx nr kolejny wysłanego SMS'a

Gdy otrzymamy SMS'a modem wyśle komunikat:

**+CMNI: „SM”,x**                    gdzie x nr komórki pamięci na karcie SIM gdzie zapisano SMS'a.

Odczytać go możemy komendą:

**AT+CMGR=x**                    gdzie x nr komórki

modem odpowie komunikatem:

**+CMGR: „REC UNREAD”,”xxxx”,”dddd,gggg”** gdzie:

**REC UNREAD**                    informacja, że SMS nie był jeszcze czytany

xxxx                                nr telefonu z którego wysłano SMS'a

dddd                                data wysłania

gggg                                godzina wysłania

Gdybyśmy wcześniej skonfigurowali komendą **AT+CNMI=2,2,0,0,0** odbieranie SMS'ów bez ich zapisywania na karcie SIM, to gdy otrzymamy SMS'a modem wyświetli jego treść w następujący sposób:

**+CMT: „xxxx”,”dddd,gggg”**                    gdzie jak w poprzednim przypadku:

xxxx                                nr telefonu z którego wysłano SMS'a

dddd                                data wysłania

gggg                                godzina wysłania

Jak widać nie ma tu informacji o statusie SMS'a.

Przeczytanego już SMS'a możemy skasować komendą:

**AT+CMGD=x**                    gdzie x nr komórki która chcemy skasować.

SMS'y możemy też odczytać komendą

**AT+CMGL="xx"**                    gdzie xx rodzaj SMS do odczytu. XX może przyjąć jedna z poniższych wartości (wielekość znaków ma znaczenie!):

**ALL**                                odczytuje wszystkie SMS'y

**REC UNREAD**                    odczytuje tylko nieprzeczytane

**REC READ**                        odczytuje przeczytane

**STO UNSENT**                    odczytuje nie wysłane

**STO SENT**                        odczytuje wysłane

Komenda **AT+CMGL="ALL"** odczyta więc wszystkie SMS'y z pamięci.

SMS'y możemy także zapisywać na karcie np. w celu późniejszego ich wysłania. Służy do tego komenda:

**AT+CMGW="xxxx"** gdzie jak w przypadku komendy +CMGS xxxx to nr telefonu. W tym przypadku także ukaże się znak zachęty. SMS'a kończymy naciskając **CTRL+Z** lub rezygnujemy przez **ESC**.

Po zapisaniu SMS'a otrzymamy komunikat:

**+CMGW: x** gdzie x nr komórki pod która SMS został zapisany.

SMS zapamiętany na karcie możemy wysłać komendą:

**AT+CMSS=x** gdzie x nr komórki pamięci z której wysyłamy SMS'a. Podobnie jak w przypadku komendy AT+CMGS tak i tu otrzymamy informacje o nr wysłanego SMS'a.

To już wszystko na temat SMS'ów. Zapomniałem tylko wspomnieć, że aby wogule wysyłać SMS'y należy wpisać nr centrum SMS. Najczęściej karta SIM, którą kupujemy od operatora ma wpisany nr centrum. Gdyby jednak tak nie było nr ten wpisujemy komendą:

**AT+CSCA="xxxx"** gdzie xxxx nr centrum. Nr ten można odczytać wydając rozkaz:

**AT+CSCA?**

Na CD-ROM znajduje się źródłowa wersja programu w Pascalu wysyłającego SMS'y.

## Połączenia audio

Modemy firmy WaveCom mają wbudowany interfejs do słuchawki i mikrofonu. Bez problemu można tam podłączyć zwykłą słuchawkę telefoniczną i wykonywać połączenia audio. Zakładam, że modem jest skonfigurowany tak jak opisałem to wcześniej (komendy +CLIP, +CMEE, +WIND, +CRC). Aby połączyć się z innym telefonem należy wydać rozkaz:

**ATDxxxx;** gdzie xxxx wybierany nr. Należy koniecznie pamiętać o średniku za nr. Średnik ten informuje, że w połączeniu które realizujemy nie będzie transmisji danych. Podczas zestawiania połączenia, podobnie jak przy transmisji danych ujrzymy komunikat:

**+WIND: 5,1**

następnie:

**+WIND: 2** oznaczający, że do abonenta żądanego jest wysyłany sygnał dzwonienia.

Gdy abonent zgłosi się otrzymamy komunikat **OK**. Połączenie można przerwać wydając rozkaz **ATH**. Jeśli przerwanie połączenia przyjdzie od abonenta lub połączenie zostanie zerwane zostaniemy o tym poinformowani komunikatem:

**NO CARRIER**

**+WIND 6,1**

Połączenie przychodzące jest sygnalizowane komunikatem:

**+CRING: VOICE**

**+CLIP: „xxxx”,145,,”yyyy”**

gdzie xxxx nr wywołujący

yyyy wpis z książki (jeśli istnieje)

Gdy abonent wywołujący przerwie wywoływanie otrzymamy komunikat:

**+WIND: 6,1**

Połączenie odbieramy rozkazem **ATA**, odrzucamy przez **ATH**.

## Obsługa książki telefonicznej

Moduł obsługuje książkę telefoniczną. Zamiast wpisywać nr wystarczy podać nazwę z książki. W przypadku komendy ATD dla połączenia audio wygląda to następująco:

**ATD>"xxxx";** gdzie xxxx wpis w książce.

Jeśli wpisu w książce nie będzie otrzymamy komunikat:

**+CME ERROR: 22**

Komenda **AT+CPBS?** informuje nas jakie książki możemy obsługiwać (na karcie SIM, w pamięci telefonu, itd.).

**AT+CPBS?** zwraca nam informacje o aktualnie obsługiwanej książce. Przykładowa odpowiedź na w/w komendę może wyglądać tak:

**+CPBS: „SM”,66,250**

oznacza ona, że książka na karcie SIM ma pojemność 250 rekordów, zapisanych jest 66.

Wyboru książki na karcie SIM dokonujemy komendą: **AT+CPBS="SM"**

Komenda **AT+CPBW=?** Poinformuje nas o tym jaką pojemność ma książka, jak długie mogą być wpisy. Przykładowa odpowiedź:  
**+CPBW: (1-250),20,(129,145),14** oznacza, że zakres wpisów w książce zawiera się w granicach 1-250, zapamiętany nr może mieć 20 cyfr, opis może zawierać 14 znaków.

Książkę odczytujemy rozkazem:

**AT+CPBR=x,y** gdzie x początkowa pozycja książki, y końcowa. Np. **AT+CPBR=1,10** wyświetli pierwsze 10 wpisów z książki, **AT+CPBR=1,250** całą książkę.

**AT+CPBW=x** gdzie x nr rekordu, kasuje wybrany wpis w książce.

**AT+CPBW=x,"yyyy",129,"zzzz"**

zapisuje dane do książki, gdzie:    x           nr zapisywanej komórki  
  yyyy       nr do zapamiętania  
  zzzz       opis

np.: **AT+CPBW=250,"+48600700800",129,"Test"**

zapisze pod 250 komórką, nr +48600700800 opisany tekstem Test. Wpis możemy odczytać **AT+CPBR=250** w wyniku czego otrzymamy:

**+CPBR: 250,"+48600700800",129,"Test"**

Możemy też szukać wpisów w książce komendą:

**AT+CPBF="xxx"** gdzie xxx szukany fragment znaków. W odpowiedzi zobaczymy wszystkie pasujące do wzorca wpisy np. w odpowiedzi na komendę:

**AT+CPBF="Ko"** otrzymamy:

**+CPBF: 10,"+48123456789",129,"Kowalski"**

**+CPBF: 23,"+48111222333",129,"Kochanski"**

Jeśli nie ma pasującego wpisu modem odpowiada:

**+CME ERROR: 22**

### **Podłączenie modemu do dowolnego procesora:**

W dziale tym opiszę sposób podłączenia i obsługi modemów wyposażonych w port RS232C oraz RS o poziomach TTL.

### **Podłączenie modemu GSM firmy WaveCom do CPU:**

Podłączenie tego modemu w wersji „FastRack” jest banalne. Modem jest wyposażony w typowy port RS232C. Z procesora musimy więc wyprowadzić linie TXD, RXD, oraz sterujące. Należy użyć konwertera poziomów z TTL na RS232C (np. dwa układy MAX232). Przykład połączenia widzimy na schemacie

**(REDAKCJA plik w protelu „Mod51\_R3.sch”).**

Modem łączymy typowym kablem modemowym ze złączami DB9pin. Programowa obsługa modemu będzie opisana w dalszych rozdziałach.

### **Podłączenie DipModemu firmy MultiTech do CPU:**

#### **Fotografia ((REDAKCJA: plik Insingth\_SLM24xx\_DIP\_Modem\_spec.pdf” str. 1)**

Na początek opiszę możliwości modemu. Modem jest zasilany napięciem z zakresu 3.3 do 3.6V (typowo 3.3V) ale na wejściach akceptuje poziomy TTL. Dzięki temu nie ma problemu z podłączeniem modemu do urządzeń zasilanych napięciem 5V. Rozkład wyprowadzeń modemu i rozmiary obudowy widzimy na rysunku **(REDAKCJA: plik Insingth\_SLM24xx\_DIP\_Modem\_spec.pdf” str. 2)**. Modem może pracować w jednym z dwóch trybów: równoległym i szeregowym. Pierwszy z trybów jest przeznaczony do podłączenia modemu do magistrali mikroprocesorowej. W drugim trybie (szeregowym) modem udostępnia nam sygnały RXD, TXD, RTS, CTS, RI, DCD. Są to typowe sygnały występujące w UART’ach. Tryb szeregowy można stosować w urządzeniach, które obsługują modem UART’em. Nie wymaga to dużej przebudowy urządzenia i oprogramowania. Jeśli jednak urządzenie projektujemy od podstaw warto zastosować tryb równoległy zwłaszcza, że mamy dostępny tam bufor FIFO (12 bajtów dla odbiorczego, 16 dla nadawczego). Dzięki wyeliminowaniu UART’a koszt urządzenia będzie mniejszy. DIP-Modem rozpoznaje typowe komendy AT oraz wiele przeznaczonych tylko i wyłącznie dla niego. Poza typowymi komendami mamy grupy przeznaczone do: dekodowania CLIP, ustawienia standardu sygnału dla danego kraju, upgrade oprogramowania i wykrywania telefonu podłączonego równoległe do linii telefonicznej.

## Tryb szeregowy

W trybie szeregowym modem łączymy z UART'em. Do komunikacji służą linie TXD i RXD. Pozostałe linie sterują przepływem danych (RTS, CTS), informują o dzwonu (linia RI) oraz pojawieniu się fali nośnej (linia DCD). Modem nie ma wyprowadzonych sygnałów DSR i DTR służących do wykrywania gotowości terminala i modemu. Nie jest to dużym problemem, ponieważ obecność modemu można wykryć wydając komendę AT, modem powinien odpowiedzieć OK. Jeśli komuś bardzo zależy na sprzętowym wykrywaniu modemu wystarczy, że wyjście wolnej bramki logicznej, na której panuje poziom niski doprowadzi na wejście DSR UART'a. Działa to dlatego, że sygnał gotowości to przeważnie nic innego jak włączenie zasilania modemu. Można też zapętląć sygnał DTR na DSR UART'a, ale wtedy nawet gdy modem nie będzie zasilany, terminal wykryje jego obecność. Najbardziej poprawnym rozwiązaniem stosowanym zresztą w wielu modemach jest zapętlenie sygnału DTR na DSR za pośrednictwem nie negujących bramek. W trybie szeregowym z DIP-Modemem komunikujemy się tak jak z każdym modemem zewnętrznym. Nie będę tu szczegółowo opisywał tego zagadnienia, ponieważ uczyniłem to przy okazji artykułu „Przystawka modemowa”.

## Tryb równoległy

W tryb równoległy „wchodzimy” podając na linie AOUT/INT poziom niski (można ściągnąć linie rezystorem  $\leq 10k$  do masy), a na linię CS wysoki podczas aktywnego poziomu na wejściu reset. Znaczenie sygnałów w trybie równoległym opisano w tablicy (**REDAKCJA: plik „SiLab\_programguide.pdf” str. 47 tab. 48**). W trybie równoległym mamy typowe sygnały magistrali Intelowskiej (RD, WR, CS, INT, A0, D0..D7). W trybie tym mamy dostęp do dwóch rejestrów. Rejestr występujący pod ofsetem 0 jest rejestrem danych. Podczas zapisu umieszczamy w nim znak do wysłania, przy odczycie modem zwraca nam odebrany znak. Pod ofsetem 1 znajduje się funkcja stanu. Znaczenie bitów tego rejestru jest następujące:

Nr bitu	nazwa	R/W	opis
Bit 7	RXF	R	0 – bufor odbiorczy ma wolne miejsca (poniżej 10 znaków w buforze) 1 - przepełniony bufor odbiorczy (ponad 10 znaków)
Bit 6	TXE	R	0 – bufor nadawczy ma wolne miejsca (poniżej 10 znaków w buforze) 1 - przepełniony bufor nadawczy (ponad 10 znaków)
Bit 5	REM	R	0 – w buforze odbiorczym są znaki 1 – pusty bufor odbiorczy
Bit 4	INTM	R/W	maskowanie przerwania
Bit 3	INT	R	rejestr przerwania 0 – nie było przerwania 1 – wystąpiło przerwanie od (CID, OCD, PRD, RI lub DCD). Bit INT kasujemy komendą AT:I, którą to można odczytać źródło przerwania. Znaczenie bitów w rejestrze U70: 5 – wykonano reset 4 – wykryto CLIP 2 – wykryto równoległy telefon 1 – wykryto dzwonek 0 – wykryto nośną modemu (DCD)
Bit 2	ESC	R/W	
Bit 1	RTS	R/W	ustawienie stanu wirtualnej linii RTS (po ustawieniu modem przerwie wysyłanie znaków do CPU)
Bit 0	CTS	R	odczyt stanu wirtualnej linii CTS (jeśli ustawiona, to modem nie może przyjmować znaków z CPU)

Uwaga!

Nie wszystkie modemy MultiTech posiadają wyprowadzoną linię umożliwiającą wprowadzenie modemu w tryb równoległy.

W trybie szeregowym podłączenie DipModemu jest jeszcze prostrze niż modemu GSM ponieważ nie są wymagane żadne konwertery z TTL na RS232C. Jest to spowodowane tym, że DipModem mam wyprowadzone wszystkie sygnały w standardzie TTL.

Dip modem MultiTechu nie był niestety wyposażony w możliwość uruchomienia trybu równoległego (nie wyprowadzono odpowiednich pinów za złącze), jakkolwiek sam układ zastosowany w modemie posiada taką możliwość.

## **Podłączenie modemu do magistrali procesora (na przykładzie przystawki modemowej)**

### **Fotografia (REDAKCJA: Model w EP)**

Sposób podłączenia modemu z RS232C do procesora omówię na przykładzie procesora 8051 oraz układu UART 16C550. UART ma tą cenną zaletę, że posiada sprzętowy bufor FIFO, dzięki czemu procesor może zawieszac przerwania a przychodzące znaki nie będą gubione.

Omówię najistotniejszy dla nas fragment przystawki czyli sposób podłączenia UART'a do procesora oraz modemu. Na rysunku

**(REDAKCJA plik protela „Mod51\_R1.sch”)**

widzimy schemat całej przystawki, na

**(REDAKCJA plik protela „Mod51\_R2.sch”)**

szczegóły dotyczące połączenia uarta z procesorem i modemem. Sygnał z dekodera adresowego uaktywnia UART linią CS\_SIO. Pozostałe linie (RD, WR, D0..7) są typowe dla mikroprocesorów Intelowskich. Jeśli UART będziemy chcieli podłączyć do procesorów rodziny Motorola należy dodać układ na bramkach NAND (np. 74HC00) wg schematu w ramce. Jest to konieczne ponieważ układy Motorolli dysponują parą sygnałów R/W i E. Wejścia/wyjścia układu UART akceptują sygnały o poziomach TTL, natomiast modem o poziomach RS232C. Do konwersji sygnałów można użyć układów MC1488/89. Jest to rozwiązanie tanie, ale układy konwerterów wymagają zasilania napięciem +/- 12V. Jeśli napięcia takie nie są dostępne można użyć konwerterów z przetwornicą MAX232. Są to układy dość popularne i tanie, ale trzeba użyć aż 3 szt. Najlepszym wyjściem wydaje się być układ MAX241 posiadający odpowiednią liczbę konwerterów. Istnieje też wersja zabezpieczona przed ESD.

Dlaczego proponuje zewnętrzny uart? Do sterowania modemem użyto układu UART (U2) typu 16C550. Podyktowane było to małą liczbą pinów procesora, aby obsłużyć linie kontrolne modemu (DTR, DSR, RTS, CTS, DCD, RI). Dzięki zastosowaniu UARTa port procesora można przeznaczyć na inne cele. UART ma jeszcze inną przyjemną cechę - posiada 16 bajtowy bufor FIRO. Może on mieć znaczenie przy dużych prędkościach transmisji. Jeśli prędkości nie będą duże można zastosować trochę tańszy układ 16C450. Program wykrywa z jakim układem ma do czynienia i ewentualnie korzysta z FIFO. Poza tym oprogramowanie realizuje programowy bufor FIFO 256bajtów. Zalety sprzętowego FIFO można odczuć jeśli do procesora podłączymy układy wyposażone w magistralę 1-Wire. Podczas transmisji po tej magistrali trzeba zawiesić przerwania, przez co bufor programowy nic nie daje. W tablicy 1 (REDAKCJA: str 13 pliku 16c550.pdf) przedstawiono rejestry układu 16C550. Różni się on od 16C450 dodatkowym rejestrem FCR obsługującym FIFO. Program może odróżnić z jakim układem ma do czynienia. Wystarczy do rejestru FCR zapisać 1. Odczyt rejestru ISR powinien dać wartość %1100xxxx (\$Cx). FIFO w uarcie może pracować w jednym z dwóch trybów: DMA lub zgodnym z 16C450. Tryb DMA jest szybszy, ale nieco bardziej skomplikowany w obsłudze. Łatwiej i prościej korzystać z trybu zgodnego z 16C450. Po włączeniu FIFO każde przyjęcie znaku wywołuje przerwanie. Jeśli procesor obsłuży je zanim przyjdzie kolejny znak nic szczególnego się nie stanie, program zakończy procedurę przerwania. Przyjście kolejnego znaku spowoduje wygenerowanie kolejnego przerwania itd. Jeśli natomiast przerwanie nie zostanie obsłużone na czas, kolejne znaki będą zapamiętane w FIFO. Gdy program przerwań odbierze jeden znak i wyjdzie z przerwania natychmiast zostanie wywołane kolejne przerwanie. W przypadku procesora 8051 wykona on jeden rozkaz programu głównego. Jak widać program obsługujący FIFO od programu bez jego obsługi różni się tylko ustawieniem odpowiedniego bitu w rejestrze FSR. Program obsługi przerwań pozostaje bez zmian. Dlatego przeróbka programu napisanego dla 16C450 na 16C550 polega na ustawieniu rejestru FSR. FIFO działa także podczas nadawania. Po wpisaniu znaku do rejestru THR uart natychmiast zgłasza przerwanie informujące o zakończeniu nadawania. Program główny sądzi, że znak już został wysłany i wpisuje kolejny do THR. Po wypełnieniu bufora kolejne przerwanie przyjdzie, gdy w buforze zwolni się miejsce. FIFO nadawcze niesie za sobą pewne niebezpieczeństwo. Gdy np. uart steruje magistralą RS485 po wysłaniu wszystkich znaków należy odłączyć nadajnik od magistrali. Pojawia się więc problem: kiedy uart skończy nadawać znaki? Wystąpienie przerwania jak widać nie jest na to dowodem. Po małym wykładzie na temat FIFO przejdźmy do tematu głównego. Do sprzęgnięcia uarta z modemem użyto układu MAX241E (U3). Układ ten konwertuje sygnały z poziomów TTL na RS232C i na odwrót. Układ jest dość kosztowny, ale ma wiele zalet (np. w budowanej przetwornicę, zabezpieczenie przed ESD). Zamiast niego można było użyć układów MC1488 i MC1489, ale wymagają one dodatkowych napięć +12 i -12V. Można też użyć trzech popularnych konwerterów MAX232, ale po uwzględnieniu dużo większej powierzchni na płycie i związanych z tym kosztów okazuje się, że cena MAX241E jest do przyjęcia. Poza tym MAX232 w standardowym wykonaniu nie dają nam ochrony przed ESD. Jeśli już jesteśmy przy UARCIE to warto wspomnieć o funkcji spełnianej przez bramkę U9E. Neguje ona sygnał przerwania. Jest to konieczne, ponieważ UART (a właściwie jego pierwowzór 8250) był zaprojektowany z myślą o PC, a tam przerwania są uruchamiane zboczem narastającym (poziom H na wyjściu UARTa).

Dane z wbudowanego w procesor portu szeregowego są konwertowane do standardu TTL w układzie U11 typu MAX232. Sygnały w standardzie RS232C są dostępne na złączu J12. Można też wykorzystać optoizolowany RS232 dostępny na złączu J6. Z drugiej strony należy zastosować układ przedstawiony na rysunku 2 (REDAKCJA: plik RYS2.JPG). Wszystkie elementy mieszczą się we wnętrzu wtyczki DB25. Do poprawnego działania konwertera należy ustawić na linii DTR napięcie +12V na RTS -12V. Jak widać izolacja jest zapewniona w obie strony dzięki obecności transoptorów na liniach odbiorczych. Gdyby połączyć dwie wtyczki DB25 z układami z rysunku 2 z komputerami mielibyśmy optoizolację optoizolowany RS232 w komputerze. Wadą takiego rozwiązania jest mała prędkość transmisji (dla CNY17 wynosi 9600b/s). Jeśli zależy nam na większej prędkości należy zastosować transoptory typu 6N137 (max 1Mb/s). Warto wspomnieć o funkcji spełnianej przez multiplexer U5 i złącza JP1 i JP2. W pierwotnej wersji przystawka miała obsługiwać takie urządzenia przez modem, które mogły być sterowane tylko za pośrednictwem portu RS. Sygnał wchodzący na złącze JP1 jest kierowany do CPU i na złącze JP2. Dzięki temu CPU może "podłuchiwać" dane pojawiające się na złączu JP1. W stanie spoczynku dane pojawiające się na złączu JP2 są za pośrednictwem multiplexera U5 przesyłane do JP1. W dowolnej chwili CPU może odłączyć sygnał przychodzący z JP2 i wysłać tam swoje dane z linii TxD. Dzięki multiplexerowi urządzenie może być połączone z komputerem czy innym urządzeniem. Na czas połączenia modemowego CPU może "przejąć" kontrolę nad sterowanym urządzeniem. Złącza JP1 i JP2 można wykorzystać do przyłączenia komputera i komórki za pośrednictwem odpowiednich translatorów poziomu napięcia. W stanie spoczynku komputer może sterować komórką, po nawiązaniu połączenia modemowego kontrolę nad komórką przejmuje przystawka modemowa. Port szeregowy nie daje dużych możliwości dlatego przystawkę wyposażono w port równoległy.

Ośmiobitowy port wyjściowy zrealizowano na układzie U20 typu 74HCT273. Zastosowano go dlatego, że posiada wejście reset. Dzięki temu po włączeniu zasilania na portach jest ściśle ustalony stan (same zera). Dane w zatrasku są zapamiętywane podczas narastającego (tylnego) zbocza na wejściu CLK. Na wejście to doprowadzono sygnał z dekodera adresowego. Sygnał z dekodera jest sumowany z sygnałem WR w bramce U21A, dzięki czemu aktywny sygnał pojawia się tylko podczas zapisu. Takie rozwiązanie daje to, że pod tym samym adresem może znajdować się port wejściowy. Nie zaleca się tu stosowania układu typu LS, które posiadają małą wydajność prądową w stanie wysokim. Najlepiej zastosować układy serii CMOS (HC, HCT, ACT). Jeśli zależy nam na dużej wydajności prądowej to można użyć układów serii F. Sygnały z portu są dostępne na złączu JP4. W modelowym urządzeniu do złącza tego podłączono diody LED, których stan można zmieniać odpowiednim rozkazem.

Port wejściowy zrealizowano na układzie U22 typu 74HCT245. Jest to dwukierunkowa brama 8-bit. Można by zastosować układ 74HCT244 (brama jednokierunkowa), ale cena tych układów jest taka sama, a przy HCT245 o wiele łatwiej zaprojektować przebieg ścieżek. Brama jest uaktywniana po podaniu stanu niskiego na wejście G układu. Sygnał ten pochodzi z dekodera adresowego i jest sumowany z sygnałem RD w bramce U21B, dzięki czemu aktywny sygnał pojawia się tylko podczas odczytu. Drabinka rezystorowa RP2 podciąga wejścia do +5V. Dzięki czemu wejściami tymi mają sterować transoptory czy styki przełącznika lub przekaźnika.

Wyświetlacz LCD jest opcjonalny (wyświetla dodatkowe komunikaty). Brak wyświetlacza LCD w niczym nie przeszkadza. Program gdy nie wykryje wyświetlacza nie zawiesi się tylko pominię procedury jego obsługi.

Wyświetlacz przyłączony jest do złącza J3. Ze względu na to, że magistrala wyświetlaczy LCD jest "Motorolowska", a magistrala procesora "Intelowska" trzeba było zastosować układ dopasowujący na U19. Magistrala "Intelowska" ma tą nieprzyjemną cechę, że są problemy z podłączeniem bufora dwukierunkowego. Przy standardowym podejściu do tego problemu sygnał strobu (zależnie od wykonania RD lub WR) zanikałby równocześnie z danymi na magistrali. Taka praca "na styk" jest dość ryzykowna. Aby uniknąć takich problemów linię RW wyświetlacza podłączono do linii A1 procesora. Spowodowało to, że adresy do zapisu i odczytu są różne, ale mamy gwarancję, że sygnał RW (sterujący kierunkiem w magistrali "Motorolowskich") ustali się przed sygnałem strobu E wyświetlacza.

Uaktywnianiem wszystkich układów peryferyjnych i pamięci RAM steruje dekodery adresowy. Zbudowano go na układzie U12 typu 74HCT138. Jest to dekodery 1 z 8 i dzieli 64kB przestrzeń adresową procesora na osiem obszarów po 8kB. Sygnały z pierwszych czterech wyjść dekodera są sumowane w bramkach U13A i U13B. Dzięki temu sygnał CS\_RAM (pin 8 U13B) jest aktywny gdy procesor zaadresuje pierwsze 32kB. Pozostałe stroby uaktywniają układ UART (\$8000-\$9FFF), wyświetlacz LCD (\$A000-\$BFFF), porty I/O (\$C000-\$DFFF). Ostatni strob (\$E000-\$FFFF) jest niewykorzystany. W tym miejscu można by zastosować układ programowany GAL. Poza dekodowaniem adresów generowałby też sygnały zapisu i odczytu do portu równoległego i sterował strobem wyświetlacza LCD. Nie zastosowałem go dlatego, że większość Czytelników nie ma dostępu do programatora układów GAL i mogliby mieć kłopoty z samodzielnym wykonaniem przystawki modemowej. Ostatnim elementem przystawki jest pamięć EEPROM (U16), w której przechowywana jest konfiguracja, kody dostępu itp. Magistrala IIC którą jest sterowany układ EEPROM jest wyprowadzona na złącze J11. Na złącze to wyprowadzone jest także napięcie zasilania i linię przerwań procesora. Dzięki czemu w razie potrzeby można przystawkę rozbudować o zegar (np. PCF8583), dodatkowe porty (np: PCF8574) czy sterownik LED (np:

SAA1064). Zaznaczę tylko, że złącze rozszerzające IIC jest identyczne jak w Rejestratorze Telefonicznym AVT-5056.

Na koniec warto wspomnieć o przyciskach SW1 i SW2. Równoczesne ich przytrzymanie podczas resetu procesora spowoduje ustawienie standardowej konfiguracji (brak kodu dostępu, liczba dzwonek autozłoszenia 3). Naciskanie ich podczas pracy nie powoduje żadnych reakcji, ale nie ma problemu, aby dodać w programie kilka linii i przyciski będą spełniały jakieś funkcje. Diody D3...D5 sygnalizują stan pracy przystawki. D3 informuje o błędach, D4 o poprawnej pracy, D5 o ustawieniu standardowej konfiguracji lub o poziomie aktywnym na optoizolowanym wejściu J7.

Dwa sygnały wyjściowe są dostępne na złączach J4 i J5. Dzięki zastosowaniu inwerterów 74LS06 wyjścia można obciążać znacznym prądem.

## Sterowanie modemem

Aby wysłać czy odebrać jakikolwiek znak do/z modemu należy odpowiednio ustawić linie sygnałowe.

- ustawiamy linię DTR w stan aktywny.

- modem powinien odpowiedzieć ustawieniem linii DSR. Jeśli sygnał ten nie pojawi się, oznacza to, że modem jest nie podłączony lub ma wyłączone zasilanie.

- ustawiamy w poziom niski linię RTS informując modem, że jesteśmy gotowi do przyjmowania danych od niego.

- aby wysłać znak do modemu sprawdzamy stan linii CTS. Jeśli jest nieaktywna oznacza to, że modem nie jest w stanie przyjąć danych i musimy poczekać aż zmieni stan linii CTS na aktywny.

- dane z modemu zapamiętujemy w buforze. Jeśli bufor jest prawie pełny ustawiamy linię RTS w stan nieaktywny. Gdy bufor opróżni się, stan linii RTS zmieniamy na aktywny zezwalając na wysyłanie kolejnych danych przez modem.

Po zainicjalizowaniu w/w linii warto do modemu wysłać komendę "ATZ" aby przywołać standardowe ustawienia. Prawie zawsze konieczne jest wysłanie jakiegokolwiek znaku (np. komendy AT) aby modem mógł określić z jaką prędkością wysyłamy do niego znaki. W przeciwnym wypadku komenda RING przysłana zostanie do UART'a ze standardową prędkością modemu, która niekoniecznie musi być taka sama jak ustawiona w UARCIe. W konsekwencji odczytamy „krzaczkę” a nie komunikat RING.

- przyjęcie dzwonka objawia się zmianą stanu linii RI i wysłaniem przez modem komunikatu "RING".

UWAGA!

Znaczenie linii CTS/RTS i DTR/DSR jest zależne od trybu pracy (simpleks/półdupleks/dupleks) i w literaturze można spotkać się z innymi opisami.

Jeśli nie mamy ochoty na zabawę z liniami CTS, RTS lub wykorzystujemy je do innego celu, modem można ustawić tak, aby ich stan był ignorowany. Odpowiadają za to komendy "Rn", "Kn" i "Dn". Przykładowa sprawdzona sekwencja ustawiająca taki tryb wygląda tak: "AT&F&R1&K0&D0W2X4"

Sprawdzone fragmenty programu wysyłającego i odbierającego znaki z/do modemu można zobaczyć na listingu poniżej.

```
=====
                org     $0000
                ljmp    start
;-----

                org     $03                ;Int0
                ljmp    irq_uart

;-----
;Inicjalizuje UART typu 16C550 (musi byc przed "init_timer")
;
init_16C550:   mov     a,quart                ;Zapisz rejestr prędkości (na podstawie "quart")
                rl     a
                rl     a
                anl    a,#%11111100
                push   ACC
                mov    dptr,#tab_quartDL
                movc   a,@a+dptr            ; odczytaj DLM do R7
                mov    R7,a
                pop    ACC                ; odczytaj DLL do R6
                inc    a
                movc   a,@a+dptr
```

```

mov R6,a

mov dptr,#LCR ;Wybierz rejestry podzielnika
mov a,#$80
movx @dptr,a
mov dptr,#DLL ; zapisz młodszy
mov a,R6
movx @dptr,a
inc dptr ; zapisz starszy
mov a,R7
movx @dptr,a
mov dptr,#LCR ;parametry transmisji
mov a,#%11
movx @dptr,a
mov dptr,#IER ;włączenie przerwań
IF UARTIRQ=1 ;----- Obsługa UARTa na IRQ -----
mov a,#%0001
ELSE ;----- Obsługa UARTa przez poling ----
mov a,#%0000
ENDI

movx @dptr,a
mov dptr,#FCR ;włączenie FIFO
mov a,#%1
movx @dptr,a
mov dptr,#MSR ;linie aktywne
mov a,#%1111
movx @dptr,a
mov wsk_rx_mod_w,#0 ;Wskaźnik bufora odbiorczego
mov wsk_rx_mod_r,#0
setb EX0 ;Zezwól na IRQ od UARTa
clr IT0 ; wyzwalane poziomem
mov dptr,#ISR ;Odczyt ISR (starawdanie typu UARTa)
movx a,@dptr
anl a,#%11000000 ; jesli bit 6 i 7 ustawiony
cjne a,#%11000000,init_16C_1
setb FL_16C550 ; to jest uart 16C550
ret
init_16C_1: clr FL_16C550 ; nie to uart 16C450
ret

```

Powyższy listing wpisuje do układu UART z tablicy „tab\_quartDL” prędkość transmisji. Ze względu na to, że pod adresem rejestrów podzielnika znajdują się także rejestry bufora nadawczego i odbiorczego odpowiednim wpisem do rejestru LCR (ustawiony bit 7) wybieramy rejestry podzielnika. Po zapisaniu ich wpisem do rejestru LCR wybieramy rejestry nadawczy i odbiorczy oraz równocześnie ustawiamy parametry transmisji 8N1. Ponadto procedura ustawia linie gotowości w stan aktywny ZAPISUJĄC ODPOWIEDNIE DANE DO REJESTRU MSR. Zależnie od opcji kompilacji włączane są przerwania zapisem do rejestru IER. Procedura uaktywnia także bufor FIFO (rejestr FCR), który w tej konfiguracji nie wymaga dodatkowej obsługi. Ustawiane są także wskaźniki bufora odbiorczego (wsk\_rx\_mod\_w i wsk\_rx\_mod\_r) oraz sprawdzany jest typ UART'a (czy posiada FIFO czy nie) przez sprawdzenie bitów 6 i 7 rejestru ISR.

#### UWAGA!

FIFO może pracować w dwóch trybach. Pierwszy tryb jest kompatybilny z układami 16C450 czy 8250. Po odebiciu bajtu po RS'ie układ ustawia bit w rejestrze statusu i (jeśli są ustawione) generuje przerwanie. Gdyby, zanim CPU odczyta bajt przyszedł kolejny znak, UART odebrał dane zostaną one zapamiętane w FIFO. Po obsłużeniu przez CPU pierwszego odebranego bajtu UART znów ustawi bit w rejestrze statusu (i ewentualnie wygeneruje przerwanie). Dzięki temu znaki nie są tracone a program obsługi odbioru znaków nie musi być modyfikowany (poza ustawieniem FIFO UART'a). Drugi tryb, tzn. DMA, pozwala na ustawienie po ilu znakach w FIFO zostanie wygenerowane przerwanie. Gdy np. ustawimy, że przerwanie ma być generowane po odebraniu 8 znaków, to po jego odebraniu musimy z uarta odczytać 8 znaków. Jako, że bufor ma 16 bajtów, to kolejne ewentualnie przychodzące znaki są gromadzone w buforze. Tryb ten sprawdza się przy szybkiej transmisji dużej ilości danych, jest natomiast bardzo kłopotliwy przy transmisji małych pakietów danych.

```

;-----
clr_buf_modem: lcall get_char_mod ;Oczyść bufor odbiorczy
jc *-3
ret

```

Powyższa procedura odczytuje wszystkie znaki z układu UART. W konsekwencji opróżnia bufor odbiorczy FIFO. Tą samą funkcję można osiągnąć ustawiając odpowiedni bit w rejestrach UART'a.

```

;-----
;Źródło przerwania określa zawartość rejestru ISR:
; Level 1: %0000 MSR - kasowanie: odczyt MSR
; ; 2: %0010 TXRDY - kasowanie: odczyt THR lub ISR
; ; 3: %0100 RXRDY - kasowanie: odczyt RHR lub ISR
; ; 4: %0110 LSR - kasowanie: odczyt ISR
; ;
irq_uart: push PSW
push ACC
push DPH
push DPL

```



```

mov  dptr,#ISR      ;Odczyt ISR pozwala określić źródło przerwania
movx a,@dptr      ;
;-----
cjne a,#%0100,irq_uart_msr ;Czy przerwanie odbiorcze?
mov  dptr,#RHR      ; tak więc odczytaj daną z UARTa
movx a,@dptr
mov  dptr,#BUF_RX_MOD ; oblicz adres w buforze
mov  DPL,wsk_rx_mod_w
movx @dptr,a      ; zapisz znak do bufora
inc  wsk_rx_mod_w  ; next adres
push ACC          ;zapamiętaj odczytany bajt
IF RTSC=1
lcall zuzycie_bufora ;Sprawdź zajętość bufora
cjne a,#255-BUF_MARG,*+3 ;czy bufor prawie pełny ?
jc   *+2+3        ; - nie to skok
lcall clr_rts      ; - tak więc powstrzymaj nadawanie modemu
ENDIF
lcall zuzycie_bufora ;Sprawdź zajętość bufora
cjne a,#255,*+3    ;czy bufor zapelniony ?
jc   *+2+3        ; - nie to skok
ljmp error        ; - tak więc mamy problem

pop  ACC          ;odtwórz zapamiętany bajt
sjmp irq_uart_exit
;-----
irq_uart_msr: cjne a,#%0000,irq_uart_isr ;Czy przerwanie od MSR?
mov  dptr,#MSR      ; tak więc kasuj przerwanie
movx a,@dptr
sjmp irq_uart_exit
;-----
irq_uart_isr: cjne a,#%0110,irq_uart_exit ;Czy przerwanie od LSR?
mov  dptr,#LSR      ; tak więc kasuj przerwanie
movx a,@dptr
sjmp irq_uart_exit
;-----
irq_uart_exit: pop  DPL
pop  DPH
pop  ACC
pop  PSW
reti

```

Powyższa procedura obsługi przerwania jest wywołana gdy w UARCIE znajdzie się choć jeden znak. Na początku określone jest źródło przerwania przez sprawdzanie rejestru ISR. Jeśli jest to przerwanie odbiorcze odczytywany jest bajt danych z rejestru RHR i zapisywany w buforze odbiorczym w mieszczącym się w pamięci RAM. Następnie sprawdzane jest zużycie bufora odbiorczego procedurą „zuzycie\_bufora”. Jeśli bufor jest prawie pełny procedura „clr\_rts” zmienia stan linii RTS informując modem aby zaprzestął przesyłania znaków do procesora. Jeśli bufor zostanie zapelniony całkowicie (co w normalnych warunkach nie może mieć miejsca) wywołana jest obsługa błędu (procedura „error”).

Jeśli przerwanie pochodzi z innego źródła niż bufora odbiorczego jest kasowane po czym procesor kończy obsługę przerwania.

```

;-----
;Oblicz różnicę pomiędzy wskaźnikami "wsk_rx_mod_w" a "wsk_rx_mod_r"
;WE: "wsk_rx_mod_w"
;   "wsk_rx_mod_r"
;WY: Acc - różnica pomiędzy wskaźnikami
;Zmienia: Acc
;
zuzycie_bufora: clr  C
mov  a,wsk_rx_mod_w
subb a,wsk_rx_mod_r
jnc  zuzycie_bufora1 ;jeśli liczba dodatnia to skocz
setb C              ;liczba ujemna to oznacza że
mov  a,wsk_rx_mod_w ; wskaźniki przewinęły się
subb a,wsk_rx_mod_r ; i trzeba obliczyć odwrotnie
zuzycie_bufora1:ret

```

Powyższa procedura sprawdza zajętość bufora odbiorczego.

```

;-----
;Wyłącz RTS (zatrzymaj nadawanie przez modem)
;WE: --
;WY: --
;Zmienia: Acc, Dptr
;
clr_rts: mov  dptr,#MCR
movx a,@dptr
clr  ACC+DTR ;RTS

```

```

movx @dptr,a
ret

```

Procedura przez wpis do rejestru MCR powstrzymuje nadawanie znaków przez modem.

```

;-----
;Włącz RTS (zezwól na nadawanie przez modem)
;WE: --
;WY: --
;Zmienia: Acc, Dptr
;
set_rts:  mov  dptr,#MCR
          movx a,@dptr
          setb ACC+DTR ;RTS
          movx @dptr,a
          ret

```

Procedura przez wpis do rejestru MCR zezwala na nadawanie znaków przez modem.

```

;-----
;Czyta znak z bufora odbiorczego
;WY: Acc - odczytany znak
; C - status (C=1 jest znak)
;Zmienia: Acc
;
get_char_mod:
IF UARTIRQ=1 ;----- Obsługa UARTa na IRQ -----
    mov  a,wsk_rx_mod_w ;Czy jest znak w buforze?
    cjne a,wsk_rx_mod_r,get_char_mod_y
    clr  C ;Znacznik - nie ma znaku
    ret
get_char_mod_y: push  DPH ; jest - rejestry na stos
                push  DPL
                IF RTSCTS=1
                    lcall zuzycie_bufora ;Sprawdź zajętość bufora
                    cjne a,#BUF_MARG,*+3 ;czy bufor prawie pusty ?
                    jnc  *+2+3 ; - nie to skok
                    lcall set_rts ; - tak więc zezwól na nadawanie modemu
                ENDI
                mov  dptr,#BUF_RX_MOD ; oblicz adres w buforze
                mov  DPL,wsk_rx_mod_r
                movx a,@dptr ; odczytaj znak z bufora
                inc  wsk_rx_mod_r ; next adres
                lcall inc_bodb ;Incrementuj licznik bajtów odebranych
                pop  DPL ; wyjdź ze znacznikiem - jest znak
                pop  DPH
                setb C
                ret
            ELSE ;----- Obsługa UARTa przez poling -----
                push  DPH
                push  DPL
                mov  dptr,#LSR ;Sprawdź czy jest cos w rejestrze
                movx a,@dptr
                anl  a,#1
                cjne a,#0,get_chat_mod1 ;-jest to skok
                pop  DPL
                pop  DPH
                clr  C ;Znacznik nie ma znaku
                ret
get_chat_mod1: mov  dptr,#RHR ;Odczytaj znak
                movx a,@dptr
                pop  DPL
                pop  DPH
                setb C ;Znacznik OK
                ret
            ENDI

```

Powyższa procedura czyta znaki z bufora odbiorczego, który jest zapisywany w przerwanu.

```

;-----
;Wpisuje znak do bufora THR i czeka na koniec transmisji
;
put_char_mod: push  ACC
                push  B

```

```

        push DPH
        push DPL
IF RTSCTS=1
        push ACC          ;Jeśli modem powstrzymuje transmisję to czekaj
        mov B,#CONST_CTS
        mov dptr,#MSR
put_char_mod2: movx a,@dptr
        jnb ACC+CTS,put_char_mod3 ; jeśli CTS nieaktywne to odliczaj czas
        sjmp put_char_mod4      ; CTS aktywne więc skocz do nadawania znaku
put_char_mod3: mov a,#1        ;Pętla czasowa
        lcall wait_10ms
        djnz B,put_char_mod2    ;jeśli nie minął czas to skok
        lcall break             ; - tak więc rozłącz modem
        mov a,#ERR_CTS
        lcall print_err
        ljmp restart           ; i restartuj system
put_char_mod4: pop ACC
ENDI

        mov dptr,#THR
        movx @dptr,a
        mov dptr,#LSR          ;Czeka na koniec transmisji
wait_thr: movx a,@dptr
        jnb Acc.5,wait_thr

        pop DPL
        pop DPH
        pop B
        pop ACC
        ret

```

Procedura wysyła znak do UARTa. Zależnie od opcji kompilacji przed wysłaniem znaku sprawdza czy modem może przyjmować dane (czy uaktywnił linie CTS) sprawdzając rejestr MSR. Jeśli modem za długo będzie powstrzymywał transmisję (np. wyłączono mu zasilanie) to zostanie wywołana procedura „break” rozłączająca połączenie oraz zasignalizowany zostanie błąd. Jeśli można wysłać znak do modemu bajt danych jest wpisywany do rejestru THR po czym procedura czeka na wysłanie znaku sprawdzając rejestr LSR. Dzięki buforowi FIFO procesorowi wydaje się, że znak został wysłany natychmiast. Nie będzie oczywiście działać tak w nieskończoność ponieważ po 16 znaku UART da sygnał, że znak nie został jeszcze nadany i procesor będzie musiał poczekać na koniec transmisji.

Opisano tylko najważniejsze fragmenty procedur obsługi UARTa. Całe listingi są dostępne na CD dołączonym do książki.

## Uruchomienie przystawki

W pierwszej kolejności sprawdzamy napięcia zasilania. Jeśli wszystkie są poprawne umieszczamy układy w podstawkach. Włączamy zasilanie i potencjometrem P1 ustawiamy kontrast wyświetlacza. Wyświetlacz nie jest konieczny do pracy przystawki, ale ułatwia procedurę uruchamiania. Podłączamy wcześniej skonfigurowany modem do przystawki (typowa konfiguracja "AT&F&R0&K3&D2W2X4"). Po włączeniu zasilania dioda D4 powinna zaświecić, a po chwili na wyświetlaczu przystawki powinny pojawić się następujące komunikaty:

```

"Modem 51 Vww"          gdzie ww-nr wersji programu
"(C)2003 by AVT"
"16C550 xxxxx"         gdzie xxxxxx-częstotliwość kwarcu UARTa
"MEM yyyy/zzkB"       gdzie yyyy-zajętość pamięci programu, zz-wielkość zainstalowanej RAM
"System Init OK"
"Ready..."

```

a dioda D4 zacznie pulsować. Jeśli pamięć konfiguracji będzie "czysta" równocześnie z D4 pulsuje D5.

Jeśli program stwierdzi jakąś nieprawidłowość pojawi się komunikat o błędzie. Każdemu komunikatowi towarzyszy odpowiednia liczba mignięć diody D3. Dzięki temu nawet jeśli do przystawki modemowej nie podłączono wyświetlacza LCD na podstawie diody D3 można wywnioskować o jaki błąd chodzi. Poniżej lista możliwych błędów:

"Err: Init Modem" (3 mignięcia D3)

Nie wykryto modemu (modem nie odpowiedział tekstem OK na komendę AT. Przyczyna może leżeć w konwerterze U3, złączu J2 kablach lub modemie. Do dalszych testów musimy zapewnić sobie dobrą widoczność diod LED modemu. Wyłączamy zasilanie przystawki. Po włączeniu zasilania DTR powinna zaświecić, a po chwili (zaraz po pojawieniu się tekstu "System Init OK" i zapaleniu diody D3) na modemie powinna mignąć dioda RX czasem opisana jako RXD lub RD. Gdy RX przestanie świecić, modem powinien odpowiedzieć mignięciem diodą TX opisaną czasem TXD lub SD. Jeśli modem odpowiada, problem leży w torze odbiorczym (od modemu do przystawki) jeśli na modemie nie zaświeca się led RX, problem leży w torze nadawczym. Brak świecenia DTR także oznacza problemy z torem nadawczym.

"Err: Stos" (4 mignięcia)

Błąd taki nie powinien pojawić się i jest spowodowany błędem w programie. W przypadku jego pojawiania się należy skontaktować się z autorem przystawki.

UWAGA! Czasem błąd ten pojawia się jeśli program nie wykryje modemu.

"ERR: Q-UART xxxx (5 mignięć D3)

Błąd ten pojawia się, gdy do UARTa dołączono kwarc o nieodpowiedniej częstotliwości lub gdy generator nie wzbudza się. W miejscu znaków xxxx pojawi się liczba szesnastkowa określająca zawartość timera mierzącego częstotliwość kwarcu. Jeśli znajduje się tam liczba 0000 oznacza to, że generator UARTa nie pracuje. Inne cyfry oznaczają wykrycie kwarcu, którego częstotliwość nie znajduje się w tablicy programu, w związku z czym nie da się zaprogramować rejestru podzielnika (rejest DLL i DLM).

"Err: XRam" (6 mignięć D3)

Błąd pojawia się, gdy procesor nie wykryje pamięci SRAM w U4. Przyczyny należy szukać w samej pamięci dekodera adresowego (U12 i U13). Przyczyna może leżeć też w nieprawidłowym funkcjonowaniu zatrasku U7.

"Err: SIO" (7 mignięć D3)

Nie wykryto układu SIO w U2. Problem może leżeć w dekodерze adresowym U12 lub generatorze układu SIO opartym o kwarc Q1. Nieprawidłowości w funkcjonowaniu zatrasku U7 należy wyeliminować, ponieważ układ SIO jest sprawdzany po pamięci RAM. Jeśli nie wykryto błędu pamięci RAM to zatrask musi pracować poprawnie.

"Err: EEPROM" (8 mignięć D3)

Nie wykryto sygnału ACK pamięci EEPROM w U16. Prawdopodobna przyczyna to uszkodzona pamięć lub nie wlutowane lub wlutowane o złych nominałach rezystory R14, R15.

"Err: Init IIC" (9 mignięć)

Nie można zainicjalizować magistrali IIC. Prawdopodobna przyczyna to zwarcie na magistrali IIC lub złe rezystory R14 i R15. Zwarcie może powodować układ podłączony do magistrali. W pierwszej kolejności należy usunąć pamięć U16 i ponowić test. Jeśli błędu nie będzie to znaczy, że powodem problemu z zainicjalizowaniem magistrali jest układ U16.

"Err: CTS" (10 mignięć)

Błąd ten raczej nie powinien pojawić się. Program monitoruje linię CTS powstrzymującą transmisję. Jeśli transmisja będzie powstrzymywana zbyt długo, pojawia się ten właśnie błąd. Teoretycznie błąd może pojawić się, gdy modem jest bardzo wolny, a program wysyła dane do modemu bardzo szybko. W obecnych czasach znalezienie wolnego modemu jest raczej niemożliwe, zwłaszcza że procesor wysyła dane z prędkością 2400bd. Błąd ten może pojawić się też, gdy włączymy modem podczas inicjalizacji przystawki.

Po pojawieniu się któregośkolwiek z wyżej wymienionych błędów, program podejmuje ponowną próbę uruchomienia przystawki. Dzięki temu chwilowe zakłócenia nie powodują zawieszenia się programu.

## **Obsługa przystawki**

Modem który chcemy używać z przystawką, należy wcześniej skonfigurować na komputerze programem terminala. Spowodowane jest to tym, że przystawka inicjalizując modem wysyła do niego komendę "ATZ". Po podłączeniu modemu do komputera i uruchomieniu programu terminala wysyłamy komendę:

"AT&F&R0&K3&D2W2X4". Powinniśmy uzyskać potwierdzenie "OK". Następnie wydajemy komendę

"AT&W". Konfiguracja komplikuje nieco proces uruchamiania przystawki, ale daje możliwość dowolnego

skonfigurowania modemu, zależnie od potrzeb. Po konfiguracji modem można przyłączyć do przystawki. W

standardowej konfiguracji modem zgłasza się po trzecim dzwonku i nie jest wymagane wprowadzenie kodu

dostępu. Standardowa konfiguracja jest sygnalizowana przez równoczesne miganie ledów D4(zielona) i

D5(żółta). Korzystając z programu terminala i dowolnego modemu, dzwoniemy na modem przyłączony do

przystawki. Po pierwszym dzwonku pojawi się komunikat "Ring...". Po odebraniu ustawionej liczby dzwońków

pojawia się napis "Modem Busy...". Jeśli w konfiguracji ustawiono zgłoszenie, po pierwszym dzwonku nie

pojawia się napis "Ring" tylko od razu "ModmBusy" . Jeśli modemy nawiążą połączenie na wyświetlaczu

ujrzymy napis "CONNECT", dioda D3 zaświeci się, natomiast do drugiego modemu zostanie wysłany tekst

"POŁĄCZENIE" oraz informacja o autorze programu. Pierwsze co powinniśmy zrobić to zmienić kod wydając

komendę "KOD". Zostaniemy wtedy poproszeni o wprowadzenie sześciocyfrowego kodu oraz jego

weryfikację. Kod podczas wpisywania nie jest widoczny. W kodzie rozróżniane są małe i wielkie litery. Jeśli

operacja zmiany kodu powiedzie się ujrzemy komunikat "Kod został zmieniony". Od teraz po nawiązaniu

połączenia będziemy proszeni o wypisanie kodu. Po 6-ciu nieudanych próbach modem rozłączy połączenie.

Wydając komendę "AUTOANSWER" możemy zmienić liczbę dzwońków po której modem zgłosi się.

Komendą "ProgOn" i "ProgOff" aktywujemy i dezaktywujemy wyjście wyprowadzone na złącze J4. "InitOn" i

"InitOff" steruje wyjściem J5. Komenda "PioOut" ustawia stan portu równoległego JP4. Wartość wpisujemy w

kodzie HEX. Stan optoizolowanego wejścia J7, równoległego wejścia JP5 oraz wyjść J4, J5, JP4 można odczytać komendą Input".

Ciekawa jest komenda "CallBack". Po jej wpisaniu wprowadzamy nr telefonu na który przystawka będzie oddzwaniać. Przystawka przerwie połączenie, po czym podejmie pięć prób oddzwonienia i nawiązania łączności z modemem. Podczas oddzwania na wyświetlaczu pojawia się napis "Dial xx yyy", gdzie xx-nr próby oddzwonienia liczony od góry, yyy-odliczany czas na nawiązanie połączenia. Gdy czas ten minie, pojawi się napis "Delay yyy", gdzie yyy-odliczany czas opóźnienia. Gdy czas dojdzie do zera, nastąpi kolejna próba połączenia się. Oddzwanianie można przerwać dzwoniąc na nr modemu przystawki. Po odebraniu dwóch dzwonek przystawka przejdzie do stanu oczekiwania, po kolejnych nastąpi zgłoszenie przystawki. Warto też wspomnieć, że wpisywanie parametru do komendy można przerwać naciskając klawisz ESC.

Komendą "CntCall" ustawiamy liczbę prób połączenia dla funkcji oddzwon.

"WaitCall" ustawia maksymalny czas na nawiązanie połączenia (od zgłoszenia modemu do pojawienia się aktywnego stanu linii DCD).

"DelCall" ustawia czas pomiędzy kolejnymi próbami dodzwonienia się.

"MemCall" umożliwia wpisanie nr, na który przystawka oddzwoni jeśli otrzyma mniej dzwonek niż ustawiono rozkazem "AutoAnswer". Dzięki temu jeśli ustawimy auto zgłoszenie po np. 3 dzwonekach, a zadzwonimy na przystawkę i po pierwszym lub drugim dzwonku rozłączymy się, przystawka oddzwoni na wcześniej zaprogramowany numer. Dzięki temu możemy inicjować połączenie na koszt linii, do której przyłączono przystawkę. Jeśli nr nie zostanie wpisany komendą "MemCall" (po i wydaniu komendy naciśniemy enter) przystawka nie będzie oddzwaniać po odebraniu dzwonka. Podobnie będzie, gdy ustawimy liczbę dzwonek równą 1.

Na rozkaz "Info" przystawka odpowiada wyświetlając ustawienia przystawki (liczba dzwonek auto zgłoszenia, czasy opóźnień dla funkcji oddzwania, itp)

Połączenie można przerwać na dwa sposoby. Można wysłać komendę "Break", wtedy przystawka przerwie połączenie. Drugi sposób to wysłać do modemu sekwencję rozłączającą modem (typowo +++ ATH). W programach terminala funkcja rozłączania jest przeważnie "podpięta" pod skrót klawiszowy. Nie jest istotne jakim sposobem rozłączymy połączenie, ponieważ programy śledzą stan linii DCD i w razie zaniku sygnału nośnej rozłączają połączenie.

Spis wszystkich komend przystawki możemy otrzymać wydając komendę "HELP" lub "?".

## **Połączenie z innym modemem**

Wysyłamy polecenie "ATDT1234567" i czekamy na komunikat CONNECT. Jeśli nie pojawi się w odpowiednim czasie (np. 40 sekund) rozłączamy się wysyłając "ATH". Jeśli tego nie uczynimy to i tak modem rozłączy połączenie po czasie ustawionym w S-rejestrach. Fakt nawiązania połączenia z drugim modemem możemy wykryć badając stan linii DCD. Jeśli jest aktywna modemy "widzą się". Zanik sygnału DCD oznacza zerwanie połączenia. Modem sam zwolni linię telefoniczną wracając do przyjmowania komend. Po wydaniu polecenia "ATDT" możemy sprawdzić czy nie otrzymaliśmy innej odpowiedzi niż CONNECT. Modem może odpowiedzieć BUSY jeśli żądany nr jest zajęty lub NO DIAL jeśli nie wykryje sygnału zgłoszenia centrali (ton 400Hz). Z wykrywaniem tonu 400Hz mogą być pewne kłopoty. Jeśli na linii do której podłączyliśmy modem włączymy jakąś usługę (np. przeniesienie numeru) to ton zgłoszenia centrali nie jest ciągły lecz co kilka sekund zanika na około 500ms. Modem może potraktować tą sytuację jako brak tonu 400Hz. Na szczęście komendą "ATX0" możemy wyłączyć oczekiwanie na sygnał zgłoszenia centrali.

Jeśli otrzymaliśmy komunikat CONNECT (może być z dodatkowymi informacjami o szybkości transmisji, protokole) możemy wysłać informacje do drugiego modemu. Wszystko co wyślemy zostanie przesłane do drugiego modemu, modem jest jakby przezroczysty i nie będzie reagował na komendy AT (nawet rozłączenie).

### **Przejdź do trybu rozkazowego:**

Aby wydać jakąś komendę podczas połączenia musimy wysłać znaki "+++". Po otrzymaniu potwierdzenia "OK" można wydać komendę np. rozłączenia "ATH". A co by się stało gdyby w ciągu znaków, które wysyłamy pojawił się ciąg "+++ATH"? Nie stało by się nic. Aby komenda została przyjęta modem po odebraniu znaków "+++" przez pewien czas nie może otrzymywać żadnych znaków. Jeśli jakkolwiek znak pojawi się zanim modem wyśle komunikat "OK", komenda nie zostanie przyjęta, modem potraktuje to jako ciąg znaków, które należy przesłać do drugiego modemu.

### **Powrót do połączenia:**

Po przejściu w tryb rozkazowy połączenie jest cały czas podtrzymywane. Ewentualne dane przychodzące do modemu są zapamiętywane w pamięci CACHE modemu. Aby powrócić do połączenia należy wydać komendę „ATO”. Po tej komendzie otrzymamy ponownie komunikat o połączeniu (CONNECT) oraz ewentualne znaki zgromadzone w CACHE modemu.

### **Przerwanie połączenia:**

Jeśli rozłączenie przyjdzie od drugiego modemu lub połączenie zostanie przerwane, modem przyśle do nas komunikat "NO CARRIER". Zamiast ciągle sprawdzać przychodzące komendy, prościej monitorować linię DCD. Stan aktywny tej linii świadczy o tym, że modemy "widzą się". Zmiana stanu linii na nieaktywny oznacza przerwanie połączenia.

Jeśli chcemy przerwać połączenie należy po przejściu w tryb rozkazowy wydać rozkaz ATH.

#### **Odbieranie połączenia:**

Można monitorować stan linii RI lub czekać na komunikat "RING". Po wykryciu dzwonka wysyłamy komendę "ATA" po czym czekamy na komunikat "CONNECT". Jeśli włączyliśmy automatyczne zgłaszanie się modemu (wartość w rejestrze S0 różna od 0) nie musimy czekać na dzwonek tylko na komunikat "CONNECT", ponieważ modem zgłosi się automatycznie i nie trzeba wysyłać komendy "ATA".

Podczas pisania oprogramowania okazało się, że w niektórych przypadkach kolejnej komendy nie można wysłać od razu po otrzymaniu komunikatu "OK", lecz trzeba odczekać pewien czas.

Cały program źródłowy realizujący powyższe zadanie można znaleźć na CD dołączonym do książki.

## **Program terminala**

Na potrzeby książki został napisany specyficzny program terminala. Charakteryzuje się on tym, że:

- umożliwia po uzyskaniu połączenia uruchomienie dowolnego programu. Przedtem jednak zwalnia port szeregowy na jego potrzeby. Po zakończeniu wywołanego programu port szeregowy ponownie zostanie zajęty.
- Umożliwia wybranie nr wewnętrznego na tle zapowiedzi DISA

Program jest napisany w DOS dla PC i może zostać wykożystany sterownikach opartych o PC.

Uruchomienie programu "AVT-Term" powoduje wysłanie do modemu pliku "Open.cfg" gdzie znajduje się łańcuch inicjalizujący modem. Jeśli program nie znajdzie pliku "Open.Cfg" uzna że to pierwsze jego uruchomienie i przejdzie do ustawiania parametrów portu szeregowego (dostępne także z menu i po naciśnięciu kombinacji Alt+U). W parametrach portu szeregowego można ustawić:

- nr portu COM (1...4)
- szybkość transmisji (600...1152000)
- sposób wybierania nr (tonowy lub impulsowy)
- liczbę bitów danych (5...8)
- liczbę bitów stopu (1 lub 2)
- parzystość (brak, parzystość, nieparzystość)
- sterowanie przepływem (brak, sprzętowe, programowe, sprzętowo-programowe)
- włączenie lub wyłączenie zapisu danych z okna do pliku (AVT-Term.log)

Użycie w plikach konfiguracyjnych (\*.cfg) sekwencji znaków "M^" powoduje wysłanie kodu enter, natomiast po sekwencji "+++" program czeka na pojawienie się sekwencji "OK", ale nie dłużej niż 2 sekundy. W menu mamy dostępne opcje nawiązującą połączenie (klawisz F1), rozłączającą połączenie (Klawisz F3) uruchamiającą program (Klawisz F2) oraz odbierającą połączenie (wysłanie sekwencji "ATA" - klawisz F4). Wybranie opcji "Połącz" powoduje ukazanie się okienka, w którym możemy wpisać wybierany nr. Jeśli modem z którym chcemy się połączyć jest podłączony do centrali telefonicznej na numerze wewnętrznym a centrala ma usługę DISA, wypełniamy dodatkowo pole "Wew.". W tym przypadku po połączeniu z centralą (słychać w głośniku zapowiedź DISY) naciskamy gadżet "Wewnętrzny". Po nawiązaniu połączenia okienko znika, jeśli natomiast połączenie nie powiodło się ujrzymy stosowny komunikat. Ta wersja programu Term ma tą przyjemną cechę, że pozwala na użycie w nr znaków specjalnych w numerze (np "!") na co niekoniecznie pozwalają inne programy (np. HyperTerm czy Term90). Rozłączenie połączenia powoduje wysłanie do modemu pliku "Close.cfg". Podobna akcja jest podejmowana przy wyjściu z programu. Przed uruchomieniem programu (klawisz F2) do modemu jest wysyłany plik "nazwa\_programu.cfg" po czym zwalniany jest port szeregowy (bez przerywania połączenia). Po powrocie z programu port szeregowy ponownie jest inicjalizowany i program "AVT-Term" przejmuje kontrolę nad portem szeregowym. Opcję uruchamiania programu posiada np. "Term90" tyle, że nie zwalnia on portu i są problemy z obsługą UARTa pollingiem. Opcje uruchamiania programu możemy przetestować uruchamiając program "UartGet". Program ten jest prostym programem terminala. Z programu wychodzimy naciskając ESC. Naciśnięcie klawisza F3 powoduje wysłanie pliku "End.cfg" rozłączającego modem. Sekwencja ta jest wysyłana także przed wyjściem z programu (Alt+X). Na koniec warto wspomnieć o możliwości zmiany sposobu wyświetlania informacji w oknie terminala kombinacją Alt+F (dane Ascii lub Hex). Jeśli komuś nie odpowiada wyżej opisany program może skorzystać z dowolnego programu terminalowego (także na innych komputerach, np.: Mac czy Amiga).

Dodatkowo na CD zamieściłem programy DOS'owe:

"UartTest" umożliwiające testowanie portów w komputerze,  
"UartInit" programowo inicjalizujący port szeregowy,  
"UartGet" - prosty program umożliwiający wysyłanie i odbieranie danych po porcie szeregowym  
oraz prosty program terminala „Coomon” napisany w VisualBasicu (umożliwia sterowanie liniami DTR/RTS).  
Do wszystkich programów dołączone są kody źródłowe

Znajdziecie tam też program „EzTerm” dla Win. Jego charakterystyczną cechą jest możliwość indywidualnego sterowania liniami DTR i RTS (podobnie jak w programie common w VB).

#### **Łączenie modemów na linii dzierżawionej**

Takie połączenie modemów może być przydatne tam gdzie trzeba transmitować dane na duże odległości ze średnią prędkością, a do dyspozycji mamy jedną parę przewodów. Inne zastosowanie to transmisja danych na duże odległości za pośrednictwem łączy dzierżawionych. Takimi łączami ze względu na występujące po drodze przejścia z torów analogowych na cyfrowe, światłowód i odwrotnie można transmitować tylko sygnały telefoniczne z pasma 300-3400Hz przez co odpada transmisja w standardzie RS485.

Jak każdy wie modemy można połączyć za pośrednictwem centrali telefonicznej. Wypróbowałem jednak inną możliwość, połączyłem modemy ze sobą kablem bez pośrednictwa jakichkolwiek urządzeń. Ze względu na to, że modem próbujący nawiązać łączność nie otrzyma tonu 400Hz, trzeba wyłączyć jego detekcję komendą „ATX0”. Wybieramy nr komendą „ATD”. Na drugim modemie musimy wykonać komendę „ATA”. Po chwili modemy „dogadają się” o czy zostaniemy poinformowani komunikatem „CONNECT”. Pojawia się pewien problem, gdy połączenie zostanie zerwane. Wtedy to jeden z modemów musi wykonać komendę „ATA”, drugi „ATD”. Ze względu na to, że modem wydający komendę „ATA” nie wie czy i kiedy drugi z modemów próbuje się z nim połączyć, musi on ciągle wysyłać rozkaz „ATA”, po czym czekać na „CONNECT” lub „NO CARRIER”. Jeśli otrzyma „CONNECT” to przechodzi do transmisji, jeśli „NO CARRIER” to ponownie wysyła „ATA”. Jeśli transmisja zostanie zerwana (najlepiej badać stan linii DCD) to ponownie wysyłamy „ATA”. Drugi modem powinien wysyłać „ATD” i tak jak pierwszy z modemów czekać na komunikat „CONNECT” lub „NO CARRIER”. Podobnie jak pierwszy modem gdy otrzyma „NO CARRIER” lub starci nośną (nieaktywna linia DCD) próbuje ponownie nawiązać połączenie wykonując rozkaz „ATD”. Po wysłaniu „ATA” (dla modemu odbierającego) lub „ATD” (dla nadającego) modem przez 60 sekund próbuje nawiązać połączenie (chyba, że zmienimy to w S-rejestrach), dzięki temu prędzej czy później modemy „dogadają się”.

### **Zawartość CD**

<b>HayesAT</b>	katalog z plikami opisującymi komendy i obsługę AT modemów analogowych, cyfrowych i GSM.
<b>Noty</b>	noty katalogowe układów UART.
<b>Programy.Pas</b>	Przykładowe programy źródłowe i wykonywalne obsługujące UART i wysyłające SMS'y.
<b>Programy.s51</b>	Program źródłowy i wynikowy przystawki modemowej.
<b>Programy.VB</b>	Program terminala napisany w VB.
<b>Schematy</b>	Schematy i rysunki płytek przystawki modemowej.
<b>Term</b>	Program terminala napisany w Pascalu wraz z kodami źródłowymi.
<b>EzTerm</b>	program terminala

Sławomir Skrzyński