Opracowanie komputerowego modelu symulatora dla łączności radioteleksowej

Piotr Krzysztof Sobolewski

Spis treści

Wykaz skrótów użytych w pracy

- ARQ: Automatic Request for Repetition – automatyczne żądanie powtórzenia
- BRS: Broadcast Receiving Station – stacja odbierająca (w systemie FEC)
- BSS: Broadcast Sending Station – stacja nadająca (w systemie FEC)
- COSPAS-SARSAT: Kosmicheskaya Sistyema Poiska Avariyknyh Sudov – kosmiczny system poszukiwania statków w niebezpieczeństwie (COSPAS) i Search and Rescue Satellite-Aided Tracking – satelitarnie wspomagany system śledzenia przy poszukiwaniu i ratowaniu
- DSC: Digital Selective Calling – cyfrowe selektywne wywołanie
- EC: Error Correction – korekcja błędów
- ED: Error Detection – detekcja błędów
- EGC: Enhanced Group Calling – rozszerzone wywołanie grupowe
- EPIRB: Emergency Position Indication Radio Beacon – radiopława wskazująca pozycję wypadku
- FEC: Forward Error Correction – wstępna korekcja błędów
- FM: Frequency Modulation – modulacja częstotliwości
- GMDSS: Global Marine Distress and Safety System – Światowy Morski System Łączności Alarmowej i Bezpieczeństwa
- NBDP: Narrow Band Direct Printing – wąskopasmowy system o wydruku bezpośrednim

- RCC: Rescue Coordination Center – ratowniczy ośrodek koordynacyjny
- SART: Search and Rescue Transponder – transponder poszukiwawczo-ratowniczy
- SSB: Single Side Band – modulacja jednowstęgowa z wytłumioną nośną
- VHF: Very High Frequency – bardzo wysoka częstotliwość, czyli UKF (ultra krótkie fale)

Wstęp

We współczesnej gospodarce morskiej szybka i niezawodna łączność ma coraz większe znaczenie. Z jednej strony ze względów bezpieczeństwa ważnym jest, żeby w sytuacji zagrożenia statek miał możliwość natychmiastowego i pewnego wezwania pomocy oraz podania swojej pozycji i rodzaju zagrożenia, z drugiej strony ze względów ekonomicznych statek powinien mieć przez cały czas podróży zapewnioną ciągłą i niezawodną łączność z armatorem.

Już od najdawniejszych czasów ludzie usiłowali zapewnić łączność statku z lądem oraz między statkami. Początkowo stosowano w tym celu sygnały świetlne i dźwiękowe, ale te mogły zapewnić łączność tylko na niewielką odległość. Od wynalezienia łączności radiowej stało się możliwe zastosowanie do celów łączności tej techniki, co dało nowe możliwości.

Obecnie łączność na morzu zorganizowana jest w GMDSS – globalny system łączności morskiej — składający sie z szeregu podsystemów, takich jak łączność satelitarna, radiotelefoniczna, oraz łączność dalekopisowa. Tematem mojej pracy inżynierskiej jest jeden z podsystemów GMDSS – łączność dalekopisowa oraz opracowany przeze mnie symulator radioteleksu.
Praca składa się z czterech rozdziałów. W rozdziale pierwszym opisuję system GMDSS (którego częścią składową jest system łączności radioteleksowej), jego założenia i podsystemy, z których się składa. Rozdział drugi jest opisem technicznych zasad działania teleksu ze szczególnym uwzględnieniem systemów ARQ i FEC. Rozdziały trzeci i czwarty opisują opracowany przeze mnie symulator radioteleksu – w rozdziale trzecim opisuję jego obsługę zaś w czwartym zasadę działania programu.
Do pracy dołączone są również załączniki. Załączniki nr 1 i 2 zawierają instrukcje do ćwiczeń „Łączność radioteleksowa w systemie ARQ i FEC" oraz „Łączność radioteleksowa ze stacją brzegową". Załącznik nr 3 zawiera algorytm programu oraz schemat pliku konfiguracyjnego oraz pliku książki adresowej. Załącznik nr 4 zawiera wybrane fragmenty kodu źródłowego programu, w załączniku nr 5 dołącznony jest CD zawierający wersję instalacyjną programu, jego kod źródłowy oraz inne materiały związane z pracą.

Rozdział 1

Opis systemu GMDSS

GMDSS – Global Maritime Distress and Safety System – jest nowoczesnym systemem zapewniającym łączność morską. Składa się on z szeregu podsystemów, działających na różnych zasadach, opierających się na różnych zasadach technicznych i w związku z tym odpowiedniejszych w różnych sytucajach. Te podsystemy składowe to:

- satelitarny, morski system radiokomnikacyjny INMARSAT
- radiopławy satelitarne – EPIRB
- satelitarny system alarmowania i lokalizacji obiektów w niebezpieczeństwie COSPAS-SARSAT
- system cyfrowego selektywnego wywołania DSC
- system łączności dalekopisowej NBDP (Narrow Band Direct Printing)
- radiotelefonia SSB w paśmie fal pośrednich i krótkich
- radiotelefonia FM w pasmie ultrakrótkofalowym
- systemy transmisji ostrzeżeń nawigacyjnych i meteorologicznych NAVTEX i NAVAREA
- satelitarny system wywołania grupowego statków EGC
- transpondery radarowe

Zadaniem systemu GMDSS jest nowoczesne, automatyczne realizowanie następujących funkcji:

- Alarmowanie

- Lokalizacja miejsca wypadku
- Łączność koordynacyjna
- Łączność na miejscu akcji
- Rozpowszechnianie informacji związanych z bezpieczeństwem
- Łączność ogólna

Alarmowanie Alarm w niebezpieczeństwie jest szybkim i skutecznym zawiadomieniem o niebezpiecznym wypadku, może być on skierowany do innych statków znajdujących się w pobliżu miejsca wypadku lub do nabrzeżnego ratowniczego ośrodka koordynacyjnego (RCC). System GMDSS umożliwia przesłanie alarmu w kierunku statek-statek, statek-brzeg oraz brzeg-statek.

Lokalizacja miejsca wypadku Umożliwieniu zlokalizowania miejsca wypadku służą w systemie GMDSS radiopławy EPIRB oraz transpondery radarowe SART, których sygnały mogą być odbierane przez pokładowe radary nawigacyjne jednostek udzielających pomocy

Łączność koordynacyjna W celu ustanowienia łączności niezbędnej do zapewnienia koordynacji działań statków i lotnictwa stosowane są urządzenia umożliwiające utrzymanie łączności pomiędzy ratowniczymi ośrodkami koordynacyjnymi i kierownikiem akcji lub koordynatorem nawodnego poszukiwania w akwenie, w którym zdarzył się wypadek.

Łączność na miejscu akcji Łączność taką utrzymuje się za pośrednictwem radiotelefonii lub radioteleksu na częstotliwościach niebezpieczeństwa w paśmie fal pośrednich i VHF

Rozpowszechnianie informacji związanych z bezpieczeństwem żeglugi Przekazywanie takich informacji odbywa się w paśmie fal średnich za pomocą radioteleksu metodą FEC w systemie NAVTEX, poprzez INMARSAT w systemie EGC, a także na falach krótkich w systemie radioteleksowym WWNWS

Aby statek spełniał wymagania GMDSS musi być wyposażony w urządzenia umożliwiające:

- wysłanie alarmu do stacji brzegowej za pomocą przynajmniej dwu niezależnych urządzeń

- odbieranie alarmu ze stacji brzegowej
- odbywanie komunikacji związanej z akcją poszukiwawczo-ratowniczą
- odbywanie komunikacji na miejscu akcji
- nadawanie oraz – jeśli jest to koniczne – odbieranie sygnałów służacych do lokalizacji
- nadawanie i odbieranie komunikatów związanych z bezpieczeństwem
- prowadzenie łączności ogólnej z systemami i sieciami lądowymi
- prowadzenie łączności statek-statek (bridge-to-bridge)

Konkretne wyposażenie statku zależy od obszaru, w którym statek operuje. System GMDSS łączy w sobie różne podsystemy składowe, z których każdy charakteryzuje się własnymi ograniczeniami co do zasięgu i rodzaju transmitowanych sygnałów. W związku z tym całość oceanów podzielono na cztery obszary:

A1 – obszar morski będący w zasięgu przynajmniej jednej stacji nadbrzeżnej ultrakrótkofalowej VHF, z którego możliwa jest ciągła i skuteczna łączność alarmowania za pomocą cyfrowego selektywnego wywołania na kanale 70 (156,525 MHz). Zasięg działania wynosi około 20-30 Mm.

A2 – obszar morski będący w zasięgu przynajmniej jednej radiotelefonicznej stacji nadbrzeżnej pośredniofalowej z wyłączeniem obszaru A1, w którym możliwa jest ciągła i skuteczna łączność alarmowania za pomocą cyfrowego selektywnego wywołania na częstotliwości 2187,5 kHz, zasięg wynosi około 150 Mm.

A3 – obszar morski z wyłączeniem A1 i A2, z którego możliwe jest ciągłe i skuteczne realizowanie alarmowania za pomocą morskiego satelitarnego systemu radiokomunikacyjnego INMARSAT pracującego w oparciu o satelity geostacjonarne.

A4 – obszar poza obszarami A1, A2 oraz A3

Rozdział 2

Rys historyczny

Już od zarania dziejów ludzie usiłowali komunikować się na odległości większe, niż umożliwia zasięg głosu; w tym celu stosowano sygnały optyczne i dźwiękowe. Plemiona pierwotne stosowały metody takie jak bębnienie, dęcie w rogi myśliwskie, sygnały dymne, ogień itp. O sygnałach świetlnych stosowanych w starożytności wspominał już w Iliadzie Homer. Kiedy w roku 1184 zdobyto Troję informację o tym do siedziby władców achajskich w Mykenach przesłano przy użyciu sygnałów ogniowych; podobnej metody w 43 r. n.e. użył cesarz rzymski Klaudiusz aby powiadomić senat Rzymu o zwycięstwie nad Brytanią. Arabowie w celu przekazywania informacji na odległość posługiwali się światłem. W średniowieczu całe wybrzeże Andaluzji, będącej niegdyś w arabskim władaniu, było usiane wieżami sygnałowymi, spełniającymi funkcję przekaźników.
Od wynalezienia w XVII w. lunety w Europie rozwinęła się sygnalizacja optyczna. 15 sierpnia 1875 Claude i Ignace Chappe uruchomili pierwsze połączenie telegraficzne na trasie Paryż – Lille. Linia miała długość 220 km i składała się z 22 stacji utrzymujących ze sobą kontakt wzrokowy za pomocą lunety. Początki elektrycznego telegrafu datują się na lata 1830-1840, kiedy wielu wynalazców w różnych krajach Europy wykorzystało elektryczność w swych aparatach. Na tym polu swych sił próbowali tacy wynalazcy, jak Paweł Szylling w Rosji, Carl Gauss i Wilhelm Weber w Niemczech, W.F. Cook i Charles Whitestone w Wielkiej Brytanii oraz Samuel Morse w Stanach Zjednoczonych. Ten ostatni, podczas podróży na statku wpadł na pomysł przesyłania sygnałów w jednoobwodowym układzie elektrycznym. W dniu 24 maja 1844 roku przez eksperymentalną linię z Waszyngtonu do Baltimore o długości 60 km przekazał on pierwszy telegram o treści „What has God wrought?”. Data ta symbolizuje narodziny nowoczesnej telekomunikacji. W Polsce pierwszą linię telegraficzną otwarto w 1852 roku - łączyła ona Warszawę ze stacją kolejową Granica.

W czerwcu 1932 r. Brytyjski Urząd Pocztowy wprowadził centrale teleksowe, początkowo tylko do użytku wewnętrznego, następnie zaś do powszechnego użytku. Wkrótce wprowadzono podobną usługę w wielu innych krajach europejskich, a w 1936 roku pierwsza sieć teleksowa połączyła Londyn i Amsterdam, a w 1937 roku również Niemcy. Do kodowania znaków używano wtedy przerywanego sygnału o częstotliwości 1500 Hz; pierwsze urządzenia teleksowe były podłączane do zwykłej linii telefonicznej, równolegle do telefonu, lecz wkrótce usługa ta stała się na tyle popularna, że miało sens używanie odrębnej sieci teleksowej. Wkrótce w Niemczech firma Siemens und Halske wprowadziła automatyczny system łączności teleksowej. W latach pięćdziesiątych międzynarodowe usługi teleksowe były dostępne w większości krajów europejskich, z Afryką można było łączyć się za pośrednictwem łączności radiowej, łączność z USA i Kanadą zapewniały kanały teleksowe pracujące na częstotliwościach akustycznych na kablu telefonicznym rozłożonym na dnie Atlantyku. Obecnie nowoczesne technologie powodują przechodzenie z powrotem na użytkowanie sieci telefonicznej.

Rozdział 3

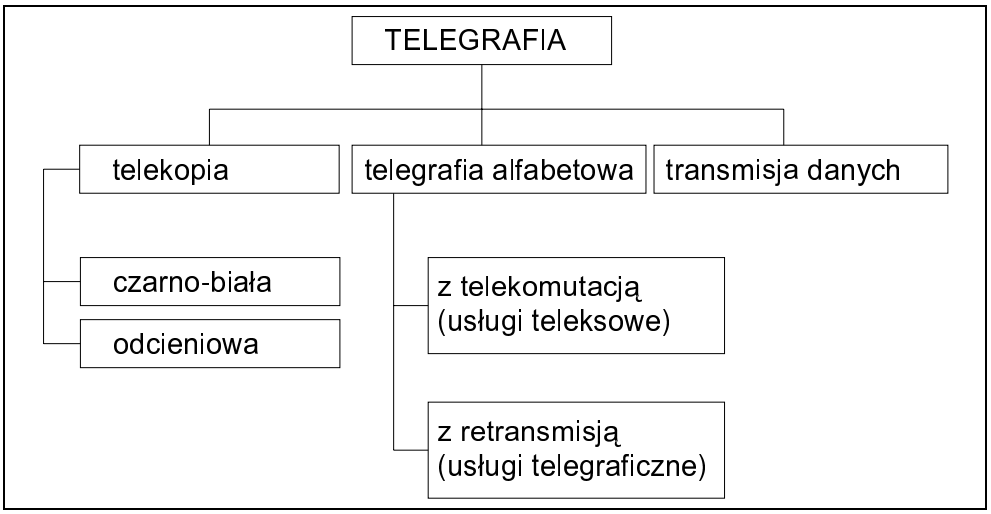
Techniczne zasady działania telegrafii

Telegrafia jest to przesyłanie na odległość obrazu, tekstu lub danych (rys 3.1). Telegrafię alfabetową (czyli zajmującą się przesyłaniem tekstu) dzielimy na telegrafię telekomutacyjną (inaczej abonencką – czyli usługi teleksowe) i retransmisyjną (czyli usługi telegramowe). Istota telegrafii retransmisyjnej polega na przekazywaniu wiadomości wysłanej przez nadawcę poprzez stacje pośrednie tak, że w końcu dociera ona do adresata. Z kolei telegrafia telekomutacyjna opiera się na złożeniu połączenia [1] między dwiema stacjami, które chcą nawiązać połączenie tak, że możliwe jest bezpośrednie przesyłanie tekstu między tymi stacjami. W systemach telegraficznych informacja przesyłana jest przy użyciu kodów telegraficznych, budowanych z elementarnych jednostek – znaków i przerw. Kody dzielimy na systematyczne – tj. takie, dla których czas trwania przerwy równy jest czasowi trwania znaku – oraz kody niesystematyczne. Ponadto kody możemy podzielić na dwustanowe i wielostanowe – kody dwustanowe posiadają tylko dwa rodzaje jednostek elementarnych, kody wielostanowe posiadają ich więcej [2]. Najstarszym historycznie kodem, który znalazł szerokie zastosowanie w telegrafii drutowej jest kod Morse'a. Jest to kod dwustanowy niesystematyczny, składający się z kropek, kresek i przerw, z których układane są dłuższe lub krótsze kombinacje oznaczające litery [3]. Stosując kod Morse'a można osiągnąć szybkość

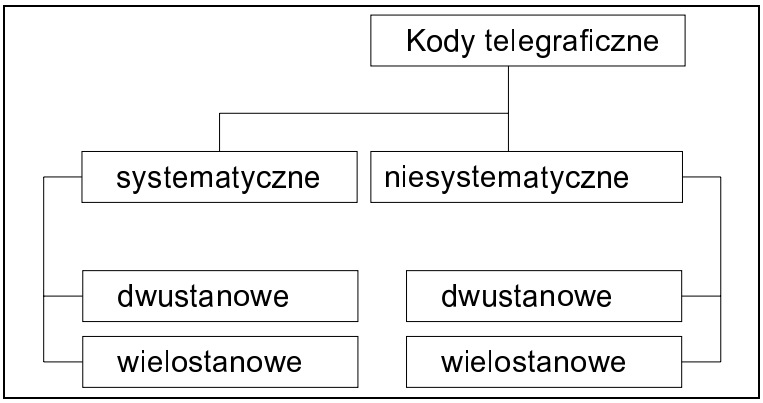
[1] Odbywa się to w podobny sposób, jak złożenie połączenia telefonicznego – tj. za pośrednictwem central, które na żądanie łączą ze sobą linie tworzące poszczególne odcinki połączenia

[2] W praktyce w radiotelegrafii nie stosuje się kodów wielostanowych, ponieważ są one bardziej od dwustanowych wrażliwe na zakłocenia – patrz rys. 3.3

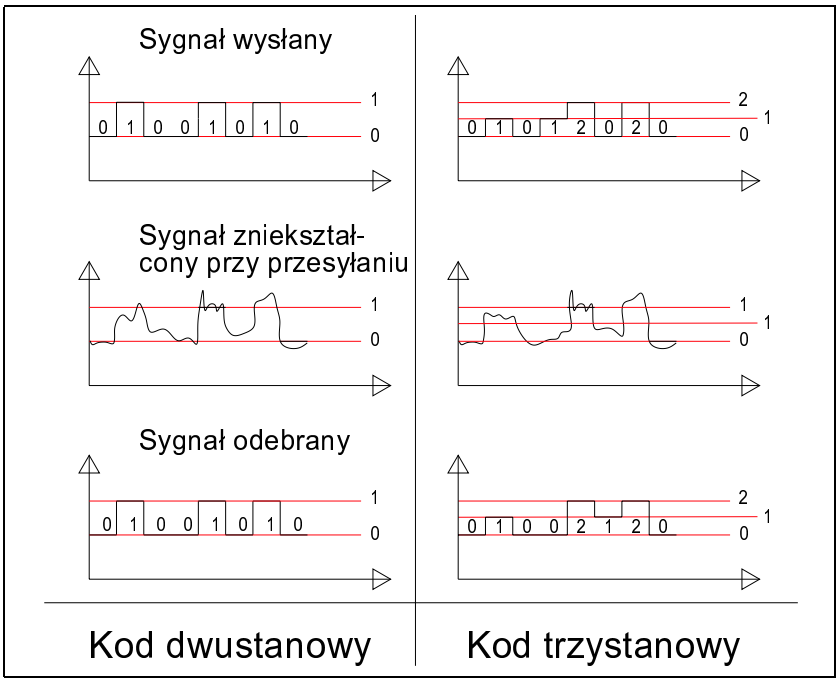
[3] Warto zwrócić uwagę że Morse, tworząc swój alfabet na sto lat przed utworzeniem podwalin teorii informacji uwzględnił częstotliwość występowania poszczególnych liter w



Rysunek 3.1: Telegrafia



Rysunek 3.2: Podział kodów telegraficznych

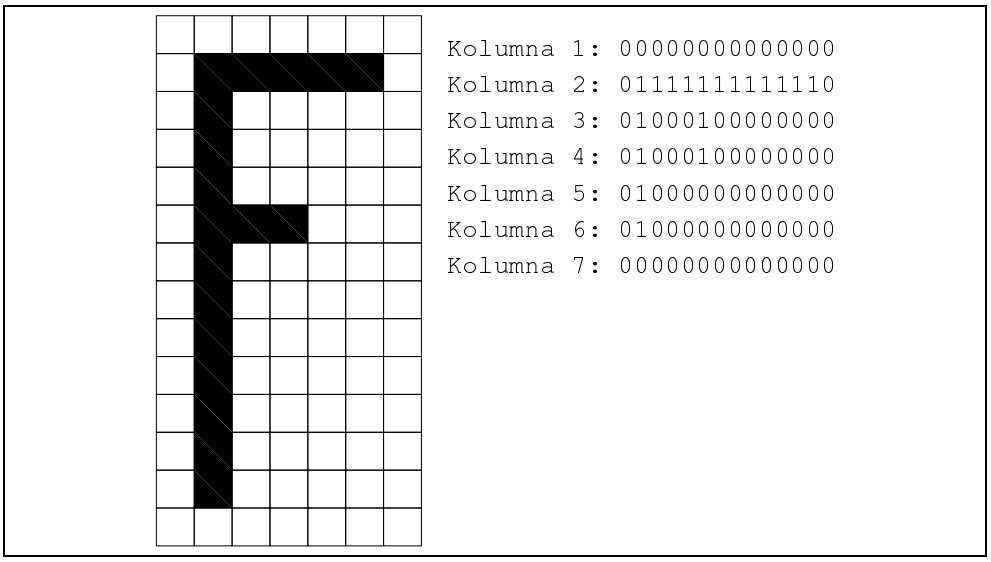


Rysunek 3.3: Wpływ zakłóceń na kody dwu- i wielostanowe

przesyłania informacji do 24 bodów [4] (bitów na sekundę) przy nadawaniu ręcznym i do 200 bodów przy nadawaniu i odbiorze automatycznym.
Obecnie w telegrafii stosuje się głównie kod pięcio- i siedmioelementowy – przy czym kod siedmioelementowy stosuje się tam, gdzie konieczna jest możliwość detekcji błędów. Oprócz tego do niedawna w niektórych służbach telegraficznych – np. w służbach prasowych – stosowano system Hella. System ten zbliżony jest do faksymilografii, polega na przesyłaniu graficznego obrazu liter w postaci ciągów znaków i przerw, którym odpowiadają elementy czarne i białe, które później złożone w prostokąt 7x14 utworzą przesłany znak (patrz rys. 3.4). Zaletą tego systemu było to, że nawet w przypadku zakłóceń tekst dało się odczytać metodą korelacji wizualnej („na oko"). Szybkość przesyłania informacji w tym systemie wynosi 122,5 boda przy nadawaniu z klawiatury i 245 bodów przy nadawaniu z nadziurkowanej taśmy.
W celu zwiększenia odporności na zakłocenia stosuje się kody z detekcją

języku angielskim w ten sposób, że częściej występującym literom odpowiadają krótsze kombinacje – w ten sposób entropia względna sygnału zakodowanego ulega zwiększeniu i w efekcie przesłanie informacji zajmuje mniej czasu.

[4]Nazwa tej jednostki pochodzi od nazwiska Emila Baudota – Francuza, wynalazcy aparatu telegraficznego wielokrotnego



Rysunek 3.4: Litera „F” i jej reprezentacja w kodzie Hella

błędów (z ang. ED – error detection) oraz korekcją błędów (EC – error correction). Możliwość detekcji i korekcji błędów można wprowadzić przez wprowadzenie w kod pewnej celowej nadmiarowości informacji (redundacji) – przez zwiększenie nierówności prawdopodobieństw poszczególnych symboli oraz zwiększanie korelacji międzysymbolowej. W przypadku kodu siedmioelementowego, stotowanego w systemach ARQ i FEC odbywa się to w ten sposób, że spośród 2^7 możliwych kombinacji znaczenie posiadają tylko te kombinacje, w przypadku których stosunek zer do jedynek wynosi 3:4 [5].

[5]Warto zwrócić uwagę na fakt, że w kod ten nie zapewnia możliwości korekcji, a jedynie możliwość detekcji błędów – ponieważ minimalna odległość między dwoma symbolami (mierzona jako ilość cyfr, którymi różnią się te symbole) wynosi 2. W ten sposób po odebraniu błędnego znaku, w którym nieprawidłowo przesłano jedną cyfrę można stwierdzić, że błąd wystąpił – ponieważ waga takiego symbolu będzie nieprawidłowa – ale nie można stwierdzić, która cyfra została przesłana błędnie.

Rozdział 4

Sposób redakcji teleksu

Cała korespondencja statkowa, w tym także teleksy wysyłane ze statku, powinny odpowiadać pewnym powszechnie przyjętym regułom, w tym ogólnym zasadom obowiązującym przy pisaniu korespondencji urzędowej. Dobry list urzędowy powinien być w miarę możliwości krótki, treściwy oraz przejrzysty i konkretny. Główna treść listu powinna być wyrażona w taki sposób, aby mogła być bez problemu zrozumiana przez każdego, kto go czyta. Pisząc list należy zwrócić uwagę na to, aby nie popełnić żadnych błędów przy podawaniu liczb, nazw własnych, oznaczeń, ilości i rodzaju towarów, czasu załadunku i wyładunku itp. Zgodnie ze zwyczajami każdy list urzędowy powinien rozpoczynać się od nagłówka zawierającego nazwę i adres firmy, numer telefonu, teleksu i faksu. Właściwa treść teleksu powinna składać się z następujących elementów:

- odniesienia, które zawiera informację o tym, w nawiązaniu do czego pisany jest teleks (np.: "W nawiązaniu do Pańskiego listu z dnia ...")
- części informacyjnej, która zawiera szczegóły i dokładniejsze informacje związane ze sprawą wspomnianą w odniesieniu
- cel listu - jest to ta część listu, w której wyjaśniamy, w jakim celu piszemy list i czego oczekujemy od adresata
- konkluzja - czyli część zamykająca list

Pisząc teleks ze względów praktycznych stosuje się wiele zwyczajowo przyjętych skrótów, niektóre z nich przedstawiam poniżej:

a – aft

abt – about

abv – above

ag – agree

adv. – advise

agt – agent

arr – arrival

asap – as soon as possible

a/s – alongside

as fllws – as follows

B/L – Bill of Lading

bred – bunkered

brg – bearing

bsh – bushel

bxs – boxes

cc – copies

c.f. – cubic feet

cfm – confirm

C.G. – Coast Guard

cgo – cargo

Ch agts – charterer's agents

c.i.f. – cost, insurance, freight

C/P – Charter Party

ctns – cartons

cts – crates

dbd – downbound

dep – departure

dft – draft

dischg – discharging

docs – documents

doz, dz – dozen

ea. – each

ER – engine room

ex – former

ex – except

flwng, flw, flwg, fllwng, flg – following

flws – as follows

f.o. – fuel oil

f.o.c. – free of charge

frm – firm *lub* from

ft – foot/feet

fw, fwd – forward

fyi – for your information

g.c. – general cargo

guar. – guaranteed

h. – hour

hcover – hatch cover

h., hf. – half

HO – heavy oil

HP – horse power

HVoy – homebound voyage

in. – inch

info – information

infod – informed

isl – island

knt, kt, kts – knot, knots

lat – latitude

lb, lbs – pound, pounds

lght – lenght

L.R. – LLoyds Register

lton – long ton

lump – lump sum

m – metre

M – mile

medaid – medical aid

mnfs – manifest

msg – message

mt – metric ton

mtlx – my telex

mty – empty

n – and

naabsa – not always afloat but safe aground

nbd – northbound

Nm – nautical miles

nn – not north of

no, nbr – number

nonflam – nonflammable

ntd – noted

oa – averall

OBO – ore/bulk/oil

obstn – obstruction

opt – option

o.t. – overtime

o'wise – otherwise

p. - per

p/bags – paper bags

pcs – pieces

pct – percent

p.o.c. – port of call

pt – point

qk – quick

qlty – quality

qty – quantity

rdr – radar

rnd, rd – round

rot – referring to our telex

rply – reply

rpm – revolutions per minute

ry – railway

rytlx – received your telex

sgnd – signed

stbd – starboard side

std – standard

sub – subject/substitute

t – ton

tdy's – today's

temp – temporary

tks – thanks

t/fore – therefore

tlx, tx – telex

tot, ttl – total

u – you/ unwatched

ull – ullage

usfl – useful

u.t. – usual terms

var – various

veg – vegetable

vegoil – vegetable oil

vert – vertical

voy – voyage

vsl – vessel

vsls – vessels

w.b. – water ballast

wbd, wbnd, wbound – westbound

wd – working day

wt – weight

yd – yard

yrs – yours

ytlx, yrtel, yrtlx – your telex

Rozdział 5

Techniczne zasady działania systemu radiotelegrafii dalekopisowej

Jak wspomniałem na początku system GMDSS składa się z wielu podsystemów składowych; w swojej pracy bliżej zająłem się jednym z nich, mianowicie łącznością dalekopisową NBDP.

Istotą działania łączności radioteleksowej, czyli wąskopasmowego systemu o łączności bezpośredniej (NBDP – narrow band direct printing) jest przesyłanie drogą radiową tekstu w postaci cyfrowej, to jest w postaci ciągu elementarnych sygnałów. Sygnały te, odpowiadające zeru i jedynce w układzie binarnym, przesyłane są jako odpowiednie częstotliwości akustyczne. Ustalone ciągi tych elementarnych sygnałów reprezentują cyfry i litery oraz znaki specjalne (takie jak znak nowej linii czy powrotu karetki), w ten sposób umożliwiając przekazanie tekstu.

W lądowych sieciach teleksowych stosuje się Międzynarodowy Alfabet Telegraficzny nr 2 (ITA No 2) – w alfabecie tym każdy znak reprezentowany jest przez ciąg pięciu sygnałów elementarnych (poprzedzonych impulsem startu o długości jednego impulsu i zakończonych impulsem stopu o długości półtorej impulsu). Alfabet ten składa się z $2^5 = 32$ różnych ciągów, spośród których 26 wybrano do reprezentowania liter, cyfr i znaków interpunkcyjnych, zaś pozostałe 6 słów spełnia funkcje sterujące. Niektóre ciągi mają dwa znaczenia – mogą one oznaczać literę lub cyfrę, czy też znak interpunkcyjny - konkretne znaczenie oznaczone jest przez odpowiedni znak sterujący (letter shift i figure shift). Standartowa szybkość transmisji w lądowych sieciach teleksowych wynosi 50 bodów (bitów na sekundę) – tak więc czas trwania jednego elementu wynosi 20 ms, a całkowity czas trwania jednego pięcioelementowego ciągu, wraz z impulsem startu i stopu – 150 ms. W ten sposób w ciągu minuty

można przesłać 400 znaków.

Alfabet ten nie nadaje się najlepiej do użytku w radiokomunikacji morskiej, ponieważ nie zapewnia on możliwości detekcji błędów. Do tego celu stosuje się Międzynarodowy Kod Telegraficzny nr 5, składający się z ciągów 7-elementowych. Istnieje $2^7 = 128$ możliwych kombinacji, spośród których wybrano do używania 35 ciągów o stosunku jedynek do zer 3:4. Dzięki temu, że wszystkie dopuszczalne ciągi mają stały stosunek jedynek do zer możliwa jest detekcja błędów i odrzucenie tych znaków, w których wykryto błąd.

Kolejna istotna różnica między tymi alfabetami jest taka, że o ile alfabet ITA no 2 przewidziany jest do pracy asynchronicznej, i w związku z tym potrzebne są sygnały startu i stopu, o tyle w kodzie siedmioelementowym, przeznaczonym do łączności synchronicznej, sygnały te nie są potrzebne. Ten opisany siedmioelementowy kod detekcyjny jest podstawą działania dwu transmisji morskiego systemu dalekopisowego: transmisji ARQ i FEC. System ARQ pracuje w zamkniętym systemie transmisji – zarówno stacja nadająca jak i odbierająca muszą być wyposażone w nadajnik i odbiornik radiokomunikacyjny. Z kolei system FEC jest systemem otwartym, rozgłoszeniowym – stacja odbierająca nie musi posiadać nadajnika.

5.1 System ARQ

System ten jest systemem synchronicznym. Łączność odbywa się poprzez dwa kanały – kanał podstawowy, służący do przesyłania właściwego tekstu oraz kanału sprzężenia zwrotnego, który służy do potwierdzenia prawidłowego odbioru lub prośby o powtórzenie błędnie odebranego bloku.

Na wyjściu dalekopisu tworzone są sygnały zgodne z alfabetem ITA nr 2, z szybkością wynoszącą 50 bodów. Następnie urządzenie ARQ konwertuje je na kod siedmioelementowy i w postaci bloków, składających się z trzech znaków, wysyła z szybkością 100 bodów. Wtedy stacja odbierająca odbiera przesłany blok, sprawdza wagę każdego z trzech znaków i kanałem sprzężenia zwrotnego wysyła informację potwierdzającą prawidłowy odbiór lub prośbę o powtórzenie bloku. W systemie ARQ używane są trzy sygnały sterująco-kontrolne: CS1, CS2 i CS3.

5.1.1 Wywołanie

Wywołanie w systemie ARQ odbywa się w ten sposób, że stacja wywołująca nadaje dwa trzyznakowe bloki zawierające numer identyfikacyjny wywoływanej stacji, w ten sposób, że jeśli numer ten to ABCD, to stacja wywołująca

nadaje:

(pierwszy blok) A RQ B (drugi blok) C D RQ

Bloki te nadawane są na przemian aż do momentu, kiedy stacja wywoływana odbierze je, rozpozna i przejdzie ze stanu gotowości (stand by) w stan odbioru, po czym nada sygnał CS1 lub CS2, które poinformują stację wywołującą, że łączność została nawiązana i że może ona rozpocząć nadawanie wiadomości.

5.1.2 Wymiana informacji

Jak już wspomniałem przesyłana informacja dzielona jest na bloki po 3 znaki. Po przesłaniu bloku stacja nadająca czeka na sygnał kontrolny nadany kanałem sprzężenia zwrotnego przez stację odbiorczą (CS1 lub CS2 – naprzemiennie). W sytuacji, jeżeli stacja odbiorcza wykryje błąd (w którymś ze znaków nie jest zachowana waga 3:4) cały blok zostaje odrzucony, a stacja odbiorcza wysyła ten sam sygnał sterująco-kontrolny co poprzednio ,co powoduje, że stacja nadawcza wysyła ponownie ten sam blok. Jeśli blok znowu nie jest odebrany prawidłowo stacja odbiorcza ponownie prosi o przesłanie go jeszcze raz i cały cykl powtarza się znowu tak długo, aż blok zostanie odebrany prawidłowo lub przekroczony zostanie limit prób. Tabela 5.1 na stronie 20 przedstawia przykładowe przesłanie słowa „RADIOTELEX" między dwiema stacjami. W systemie ARQ podczas łączności między dwiema stacjami jedna z nich – ta która akurat nadaje – jest stacją nadającą, ISS; zaś druga stacja – ta która właśnie odbiera jest stacją odbierającą – IRS. Oprócz tego stacja, która zainicjowała łączność przez cały czas aż do rozłączenia, niezależnie od tego w którą stronę odbywa się łączność, jest stacją główną – master, zaś druga stacja jest stacją podległą – slave. Funkcje te (master/ slave) związane są z synchronizacją. Idea synchronizacji przy połączeniu w systemie ARQ polega na tym, że stacja główna (master) narzuca rytm procesu synchronizacyjnego taktowanego swoim własnym zegarem stacji odbiorczej, która dostosowuje się do sygnałów odbieranych ze stacji głównej.

5.1.3 Zmiana kierunku nadawania

Zmiana kierunku nadawania może być inicjowana zarówno przez IRS jak i przez ISS. Kiedy stacja ISS chce zmienić kierunek łączności to niezależnie od tego, czy jest stacją główną, czy też podległą odbywa się to w ten sam sposób: stacja ISS wysyła stacji IRS sygnał „over" lub „+?". IRS po odebraniu tych znaków emituje sygnał kontrolny – odpowiednio CS1 lub CS2 (taki sam jak

Stacja nadawcza (ISS)	Stacja odbiorcza (IRS)
nadaje blok „RAD”	odbiera prawidłowo blok „RAD” i odpowiada „CS1”
nadaje blok „IOT”	odbiera prawidłowo blok „IOT” i odpowiada „CS2”
nadaje plok „ELE”	odbiera blok i wykrywa, że jeden ze znaków został odebrany nieprawidłowo (ma nieprawidłową wagę). Blok zostaje odrzucony a IRS ponownie odpowiada (CS2)
ponownie nadaje blok „ELE”	odbiera prawidłowo blok „ELE” i odpowiada „CS1”
nadaje blok „X ”	odbiera prawidłowo blok „X ” i odpowiada „CS2”
...	...

Tabela 5.1: Przykład przesłania tekstu w systemie ARQ

po odebraniu każdego innego bloku), na co ISS odpowiada blokiem składającym się z sygnałów „*idleβ*". Wtedy IRS nadaje sygnał kontrolny CS3, na co ISS nadaje *(βαβ) (RQ RQ RQ)*. Po odebraniu pierwszego sygnału RQ stacja ISS zmienia swój status na IRS, a po drugim nadaje sygnał CS1, po czym rozpoczyna zwykłe nadawanie trzyznakowych bloków jako IRS. Z kolei kiedy to stacja odbierająca (IRS) chce zainicjować zmianę kierunku nadawania to, po wciśnięciu przez użytkownika przycisku „over" lub CTRL O wysyła ona sygnał sterująco-kontrolny CS1, po odebraniu którego ISS rozpoczyna opisaną już procedurę zmiany kierunku nadawania.

5.1.4 zakończenie komunikacji

Kiedy stacja ISS chce zakończyć połączenie to nadaje blok składający się z trzech znaków "idle α", po czym przechodzi w stan standby. Również stacja IRS po odebraniu tego bloku przechodzi w stan gotowości. Podstawową zaletą tego systemu jest możliwość bardzo efektywnej korekcji błędów; dzięki potwierdzaniu prawidłowego odebrania każdego bloku istnieje możliwość, aby w razie potrzeby transmitować go ponownie aż do prawidłowego go odebrania. Wadą tego systemu jest konieczność wyposażenia stacji odnierającej w nadajnik; ponadto tego systemu nie można stosować w celu nadawania wiadomości do wielu odbiorców.

5.2 System FEC (kolektywny)

Ogólna idea tego systemu polega na tym, że stacja nadająca – BSS - synchronicznie nadaje strumień sygnałów, który może być odbierany przez jedną lub więcej stację BRS. Charakterystyczne dla tego systemu jest to, że łączność odbywa się tylko w jedną stronę, bez potwierdzania odbioru przez stację odbiorczą. W celu podniesienia jakości transmisji wszystkie znaki nadawane są dwukrotnie – najpierw raz, w tak zwanej transmisji DX, a później - z opóźnieniem 280 ms – drugi raz, w transmisji RX.

5.2.1 Nawiązanie łączności

Nawiązanie łączności w kolektywnym systemie FEC odbywa się w ten sposób, że BSS nadaje przemiennie sygnały fazujące a (w transmisji DX) i RQ (w transmisji RX). Po odebraniu takiego ciągu stacja przechodzi ze stanu "standby" w stan CFEC i na podstawie sygnałów fazujących 1 i 2 (a i RQ) określa która z transmisji jest transmisją DX a która RX.

5.2.2 Korekcja błędów

Stacja BRS odbiera znaki nadawane w transmisji DX i bada ich wagę. Jeśli waga któregoś znaku okaże się nieprawidłowa wtedy BRS próbuje odebrać go w transmisji RX. Jeśli i to się nie uda wtedy wydrukowany jest symbol błedu - na przykład * lub spacja.

5.2.3 Zakończenie połączenia

Zakończenie łączności odbywa się w ten sposób, że stacja BSS nadaje po kolei trzy sygnały „a" w transmisji DX i kończy nadawanie. Z kolei stacja BRS po odebraniu przynajmniej dwu sygnałów a w transmisji DX odczekuje przynajmniej 210 ms, a nastepnie przechodzi w stan „standby".

5.3 System FEC (selektywny – SFEC)

Jest to odmiana systemu FEC umożliwiająca nadawanie tylko do jednego, konkretnego odbiorcy. Możliwe jest to dzięki temu, że przy wywoływaniu, po nadaniu sygnałów fazujących stacja nadająca nadaje czterocyfrowy numer stacji. Inną różnicą w stosunku do kolektywnego FEC jest odwrotny stosunek jedynek do zer.

Podstawową zaletą systemu FEC jest fakt, że stacja odbierająca nie musi posiadać nadajnika. Poza tym w systemie tym istnieje możliwość nadawania wiadomości skierowanych do więcej niż jednego odbiorcy – takich, jak prognozy pogody, ostrzeżenia nawigacyjne itp. Wadą tego systemu jest mniej efektywna niż w ARQ korekcja błędów.

Rozdział 6

Opis obsługi symulatora radioteleksu

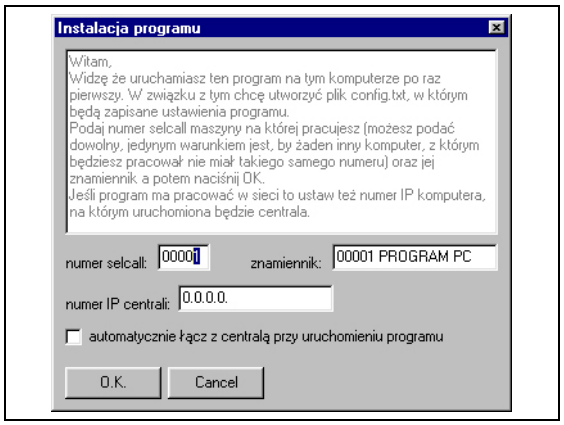
Program "Symulator radioteleksu" jest symulatorem urządzenia radioteleksowego. Ma on służyć do celów edukacyjnych, tj. do ćwiczenia obsługi radioteleksu – w tym łączenia się ze stacjami brzegowymi oraz łączności teleks-teleks. Wymagania sprzętowe to:

- komputer PC z procesorem 486 lub lepszym
- 32 MB RAM-u lub więcej

Program powinien być uruchamiany pod systemem Windows 95 lub 98, prawdopodobnie będzie działał również pod Windows NT. W celu wykonywania łączności teleks-teleks komputer powinien być podłączony do sieci. Program został napisany w języku Object Pascal.

Symulator teleksu nie jest symulatorem żadnego konkretnego odbiornika radioteleksowego, jego celem jest odtworzenie podstawowych funkcji dostępnych w większości urządzeń produkowanych przez różne firmy tak, aby osoba która nauczy się obsługi radioteleksu na tym symulatorze znała ogólne zasady pracy z teleksem i umiała obsłużyć każde urządzenie tego typu. Symulator teleksu spełnia trzy główne funkcje:

- umożliwia podstawową obsługę odbiornika teleksowego, to jest przygotowywanie wiadomości we wbudowanym edytorze tekstu, korzystanie z książki adresowej itp.
- umożliwia nawiązanie łączności z zasymulowaną stacją brzegową, w tym obsługę podstawowych komend związanych z wysyłaniem i odbieraniem telegramów, odbieraniem prognoz pogody, wysyłaniem depesz AMVER itp.



Rysunek 6.1: Okno instalacji programu

- nawiązanie przez Internet łączności z innymi użytkownikami programu w trybie ARQ lub FEC.

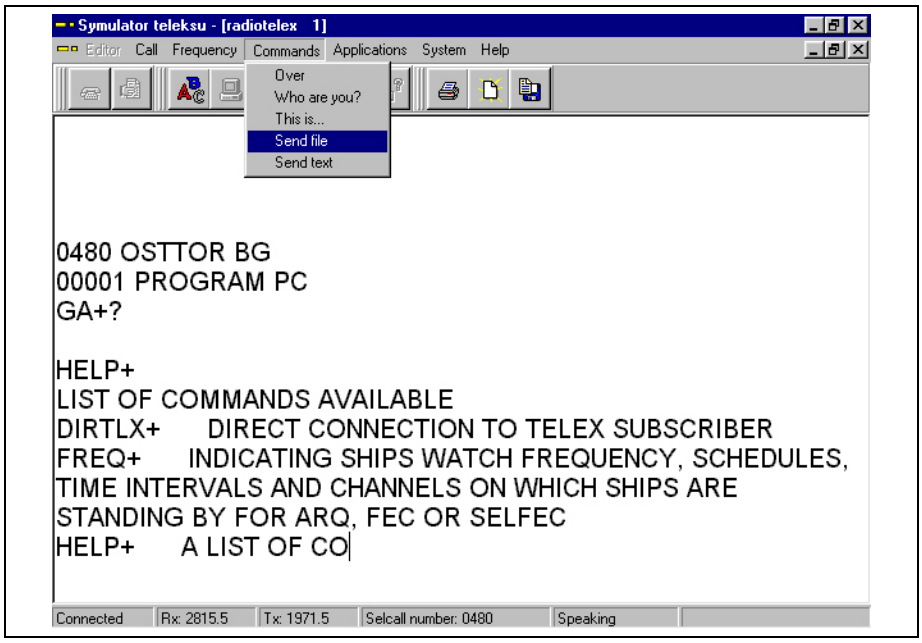
6.1 Instalacja i deinstalacja

W celu zainstalowania programu należy uruchomić program „setup.exe". Przy pierwszym uruchomieniu programu na danym komputerze (rysunek 6.1) program pyta użytkownika o numer selcall który ma być przyznany temu komputerowi oraz numer IP komputera, na którym uruchamiana będzie centrala (program niezbędny do pracy symulatora w sieci) po czym tworzy na dysku C katalog "telex" w którym zapisuje plik z ustawieniami programu pod nazwą "config.txt" oraz plik książki adresowej "adresy.txt". Ponieważ ustawienia zapisane są w postaci pliku tekstowego więc możliwe jest proste ich zmienianie po prostu poprzez edytowanie pliku config.txt.

Deinstalacja programu sprowadza się do wykasowania programu z dysku oraz skasowanie katalogu c:/telex/ .

6.2 Praca z symulatorem teleksu

Program widziany od strony użytkownika składa się z trzech podstawowych części: głównego okna radioteleksu, prostego edytora tekstu oraz książki adresowej. Po uruchomieniu programu użytkownik znajduje się w głównym oknie radioteleksu (rysunek 6.2). Na górnej belce znajduje się napis *„Symulator radioteleksu – [radiotelex 1]"* - informuje on, że znajdujemy się w głównym oknie radioteleksu o numerze selcall 0001. W górnej części okna znajduje się

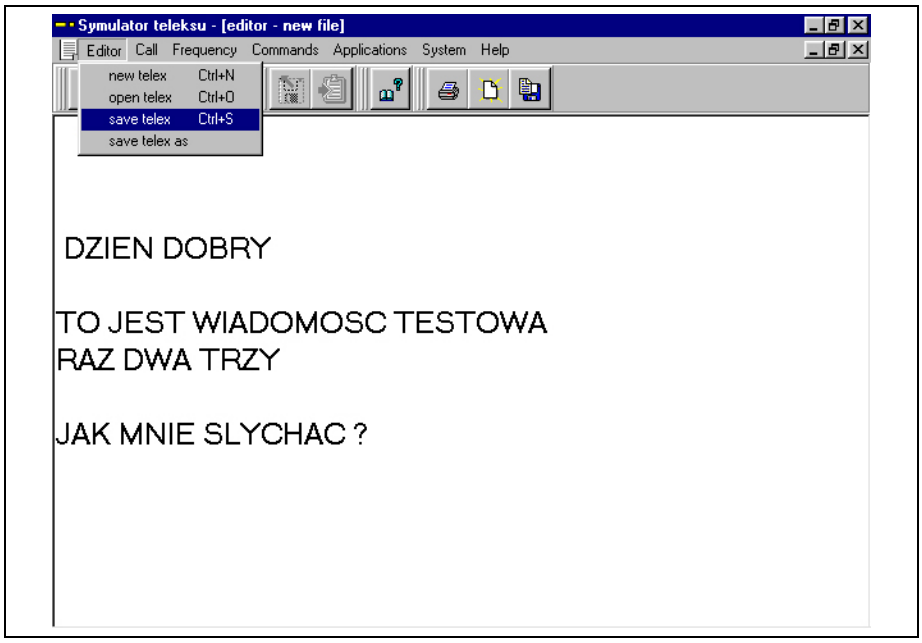


Rysunek 6.2: Główne okno radioteleksu

menu, z którego można wybierać większość podstawowych funkcji, poniżej znajduje się pasek z przyciskami. Elementy menu oraz przyciski odpowiadające funkcjom, które w danej chwili są niedostępne (np. tym, które dostępne są tylko w edytorze tekstu lub które wymagają połączenia ze stacją brzegową lub innym użytkownikiem) są szare i nieaktywne. Po najechaniu kursorem na przyciski i większość innych elementów interfejsu użytkownika pojawiają się podpowiedzi, wyjaśniające ich funkcję. Z okna radioteleksu można przejść do edytora tekstu – przez wybranie z menu „Applications – Editor" lub wciśnięcie przycisku „Edytor" (trzeci z lewej).

Po wejściu w okno radioteleksu napis na górnej belce aplikacji zmienia się na „Symulator radioteleksu – [editor – new file]" co oznacza, że jesteśmy w oknie edytora tekstu i że edytujemy nowy plik, któremu nie nadaliśmy jeszcze żadnej nazwy. Aby teraz powrócić do głównego okna radioteleksu należy z menu wybrać „Applications – Radiotelex" lub wcisnąć przycisk „Radioteleks" (czwarty z lewej).

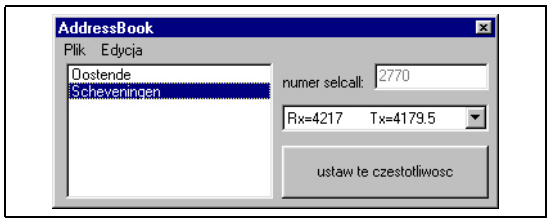
Trzecim głównym elementem programu jest książka adresowa. Aby otworzyć jej okno należy z menu wybrać „Applications – Address book" lub wcisnąć przycisk „Książka adresowa" (czwarty z prawej).



Rysunek 6.3: Okno edytora tekstu

6.2.1 Edytor tekstu

Edytor tekstu służy do tworzenia, zapisywania i odczytywania z dysku wiadomości przesyłanych teleksem. Jego obsługa jest prosta i bardzo podobna do typowych prostych edytorów tekstu takich jak np. Notatnik (Notepad). Po wejściu w okno edytora tekstu widzimy ekran taki jak na rysunku 6.3 Większą część ekranu zajmuje biała "kartka" na której wpisujemy nasz edytowany tekst. Wprowadzać z klawiatury można jedynie znaki zawarte w alfabecie ITA no 5, z tego powodu nie można wprowadzać małych liter (alfabet ITA 5 nie przewiduje rozróżnienia między małymi i dużymi literami) oraz niektórych symboli specjalnych; niedostępne są też tzw. „polskie znaki", czyli takie jak „ą" czy „ę". Po wpisaniu treści wiadomości można zapisać ją na dysku wybierając z menu polecenie „Editor – Save telex as". Po wybraniu tego polecenia program pyta o nazwę pliku, która od tej pory pojawi się w górnej belce aplikacji. Później, kiedy dokument ma już nadaną nazwę żeby go zapisać pod tą nazwą wystarczy wybrać „Editor – Save telex", żeby program zapisał dokument nie pytając drugi raz o nazwę. Jeżeli chcemy otworzyć dokument zapisany na dysku należy wybrać polecenie „Editor – Open telex", z kolei aby wyczyścić okno edytora i rozpocząć pisanie nowego tekstu należy wybrać „Editor – New telex"



Rysunek 6.4: Okno ksiazki adresowej

6.2.2 Książka adresowa

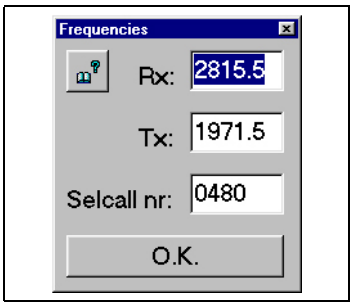
Okno książki adresowej wygląda tak, jak na rysunku 6.4. Po lewej stronie okna książki adresowej znajduje się lista stacji brzegowych zapisanych w książce. Każdej ze stacji przypisany jest numer selcall oraz pewna ilość par częstotliwości Rx i Tx. Po wybraniu stacji z listy po prawej stronie w okienku po prawej stronie ukazuje się przypisany tej stacji numer selcall, zaś w okienku poniżej jedna z par częstotliwości na których pracuje ta stacja. Aby wybrać inną parę częstotliwości należy kliknąć na przycisk z czarnym trójkątem znajdujący się z prawej strony okienka – wtedy lista częstotliwości rozwinie się i będzie można wybrać jedną z dostępnych par częstotliwości. Po wybraniu stacji i częstotliwości można kliknąć przycisk „ustaw tę częstotliwość" co spowoduje, że wybrana para częstotliwości oraz numer selcall zostaną ustawione i będą użyte przy następnej próbie łączenia się.

Do edytowania książki adresowej służą polecenia z menu „Edycja".

Usuń stację – to polecenie powoduje usunięcie wybranej stacji z książki adresowej, wraz z ustawieniami numeru selcall i częstotliwości Rx i Tx.

Dodaj stację – to polecenie powoduje dodanie do książki adresowej nowej stacji, która początkowo ma przypisany numer selcall 8888, nazwę „nowa stacja" i nie ma przypisanej żadnej pary częstotliwości. Po dodaniu do książki adresowej nowej stacji należy przy użyciu poleceń „zmień nazwę", „zmień numer selcall" i „dodaj częstotliwość" ustawić właściwe parametry stacji.

Po dodaniu nowych stacji można zapisać książkę adresową w postaci pliku tekstowego za pomocą poleceń „Otwórz" i „Zapisz" menu z „Plik". Przy każdym uruchomieniu programu program odczytuje książkę adresową zapisaną w katalogu „C:/telex" pod nazwą „adresy.txt" – tak więc jeśli chcemy, żeby nasza książka była odczytywana domyślnie przy uruchamianiu programu należy ją zapisać w tym katalogu pod tą nazwą.



Rysunek 6.5: Okno służące do wyboru częstotliwości

6.3 Główne okno radioteleksu

Po uruchomieniu programu użytkownik znajduje się w głównym oknie radioteleksu; wygląd tego okna przedstawiony jest na rysunku 6.2. Kiedy użytkownik znajduje się w tym oknie może łączyć się ze stacjami brzegowymi lub, jeśli pracuje w sieci, z innymi użytkownikami symulatora.

Zarówno podczas pracy ze stacją brzegową, jak też podczas łączności z innymi użytkownikami przydatna może być możliwość zapisania zawartości okna na dyskietkę w postaci pliku tekstowego – na przykład po to, żeby jakiś fragment tego, co się robiło dołączyć do sprawozdania. Do tego celu służy przycisk „Zapisz zawartość okna radioteleksu" – pierwszy z prawej. Jeśli użytkownik chce mieć wydruk swojej pracy z radioteleksem może użyć przycisku „Włącz/ wyłącz drukowanie" – kiedy ten przycisk jest wciśnięty wszystko, co jest wyświetlane na ekranie będzie również drukowane na drukarce – podobnie jak dzieje się to w zwykłym teleksie. Podczas pracy ze stacją brzegową przydatna bywa możliwość wysłania uprzednio przygotowanego tekstu; na przykład przygotowanego do wysłania telegramu lub depeszy AMVER. Do tego celu służą przyciski „Wyślij plik" i „Wyślij tekst (z edytora)" oraz odpowiadające im polecenia z menu „Commands" – „Send text" i „Send file".

6.3.1 Łączenie ze stacją brzegową

Aby połączyć się ze stacją brzegową należy:

- wybrać odpowiednią częstotliwość
- z menu "Call" wybrać polecenie "ARQ" (lub wcisnąć przycisk "ARQ" - pierwszy z lewej)

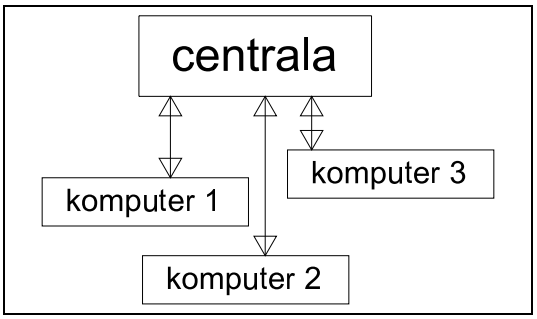
BRK+	Rozłączenie się
DIRTLX+	Bezpośrednie połączenie z abonentem brzegowym
NAV+	Żądanie przesłania ostrzeżeń nawigacyjnych
WX+	Żądanie przesłania prognozy pogody
TLX+	Wysłanie teleksu
TGM+	Wysłanie telegramu
AMV+	Wysłanie depeszy AMVER

Tabela 6.1: Lista komend

Po wykonaniu tych czynności pojawia się napis „GA+", co oznacza, że połączenie zostało nawiązane i można wydawać komendy. Spis poszczególnych komend oraz ich znaczenie przedstawia tabela nr 6.1 na stronie 29 Po zakończeniu komunikacji ze stacją brzegową należy się rozłączyć wpisując z klawiatury „BRK+" lub wybierając z menu „Commands" polecenie „Disconnect". Są dwa sposoby ustawiania częstotliwości i numeru selcall: można wejść w menu „Frequency" i wybrać dowolne częstotliwości Rx i Tx oraz numer selcall lub też wejść w książkę adresową i po wybraniu żądanej stacji i pary częstotliwości wcisnąć przycisk „Ustaw tę częstotliwość". Po połączeniu się z wybraną stacją brzegową można wydawać komendy. Listę komend dostępnych dla danej stacji wraz z krótkim opisem można otrzymać wpisując komendę „HELP+".

6.3.2 Praca w sieci

Idea realizacji połączenia między użytkownikami pracującymi na różnych komputerach przedstawiona jest na rysunku 6.6. Jak widać cała łączność między poszczególnymi komputerami przechodzi przez komputer, na którym uruchomiony jest dodatkowy program „centrala.exe". Jeśli komputer nr 1 chce przesłać informację do komputera nr 2 nie śle on jej bezpośrednio lecz



Rysunek 6.6: Połączenie między symulatorami uruchomionymi na różnych komputerach

wysyła ją do komputera-centrali, który dopiero przesyła ją komputerowi nr 2. Dlatego też aby poszczególne komputery z uruchomionym symulatorem teleksu mogły się ze sobą komunikować wszystkiem te komputery muszą być podłączone do intenetu, a poszczególne symulatory muszą być podłączone do centrali; ponadto każdy z nich musi mieć inny numer selcall. Tak jak wspomniałem w poprzednim rozdziale przy instalacji program pyta o:

- numer selcall – należy wybrać unikalny numer dla każdego komputera
- numer IP komputera, na którym uruchomiony jest program „centrala.exe"

Ponadto można zaznaczyć opcję „Automatycznie łącz z centralą przy uruchamianiu programu" – spowoduje to, że przy każdym uruchomieniu symulatora program łączy się z centralą. Jeśli przy instalacji programu nie podamy odpowiednich danych można je później zmienić przez ręczną edycję pliku konfiguracyjnego. Jeśli więc chcemy pracować w sieci należy:

- na jednym komputerze uruchomić program „centrala.exe"
- na pozostałych komputerach uruchomić symulator teleksu, podając mu (np. w pliku konfiguracyjnym) numer IP komputera, na którym uruchomiono „centralę"
- we wszystkich symulatorach teleksu włączyć z menu „System" funkcję „Łącz z centralą".

Teraz, kiedy już wszystkie symulatory są podłączone do centrali można nawiązać łączność między poszczególnymi komputerami.

ARQ

Aby nawiązać z innym symulatorem łączność w systemie ARQ należy:

- na obu komputerach ustawić (przez menu "Frequency") odpowiadające sobie częstotliwości Rx i Tx
- na komputerze wywołującym ustawić numer selcall komputera wywoływanego (wchodząc w ustawianie częstotliwości przez menu "Frequency" i wpisanie odpowiedniej wartości w okienko "numer selcall")
- na komputerze wywołującym wcisnąć przycisk "ARQ" lub wybrać z menu "Call" opcję "ARQ"

Po wykonaniu tych czynności oba symulatory wymieniają między sobą znamienniki i łączność zostaje nawiązana. Aby przesłać z jednego komputera na drugi wiadomość należy wpisać ją z klawiatury komputera nadającego, a następnie wpisać "+?" lub z menu "Commands" wybrać "Over" aby zmienić kierunek nadawania. Wpisany tekst zostanie wtedy przesłany do drugiego komputera i zostanie wyświetlony na jego ekranie. Jeśli chcemy się rozłączyć należy wybrać z menu komendę "Disconnect"

FEC

Aby nawiązać z innym symulatorem łączność w systemie FEC należy:

- na komputerze nadającym jako częstotliwość Tx (nadawania) wpisać częstotliwość Rx (odbioru) komputera (lub komputerów) do których chcemy nadawać
- w edytorze przygotować tekst, który chcemy nadać
- wcisnąć przycisk "FEC" lub wybrać z menu "Call" opcję "FEC"

Po wykonaniu tych czynności tekst z edytora zostanie nadany, po czym odebrany i wyświetlony przez wszystkie komputery odbierające na częstotliwości Tx komputera nadającego.

6.3.3 Plik konfiguracyjny

Z założenia zwykły użytkownik nie ma potrzeby edytowania pliku konfiguracyjnego (znajdującego się w katalogu „C:/TELEX/"), jednak prowadzący ćwiczenia może czasem potrzebować dokonać pewnych zmian w tym pliku tak, by zmienić ustawienia programu. W przypadku, kiedy w pliku tym dokona się nieprawidłowych zmian może zdarzyć się, że program nie będzie

działać lub też będzie się zachowywać w nieprzewidziany sposób – w takiej sytuacji najprościej jest po prostu skasować katalog „C:/TELEX/", a wtedy program przy pierwszym uruchomieniu odtworzy plik konfiguracyjny z domyślnymi ustawieniami. Pilk „config.txt" składa się z sekcji, z których każda rozpoczyna się linią składającą się ze znaku „#" i nazwy sekcji; po tej linii następują linie z treścią sekcji. Sekcje mogą być ułożone w dowolnej kolejności, ważne jest jedynie aby żadnej z sekcji nie brakowało i aby plik kończył się linią zawierającą tylko znak „#". Plik konfiguracyjny składa się z następujących sekcji:

- moj numer selcall – sekcja ta zawiera numer selcall komputera
- moj znamiennik – ta sekcja zawiera znamiennik komputera
- zwolnienie – ta sekcja określa wartość od której zależy szybkość pracy symulatora – m.in. szybkość wypisywania tekstu na ekran, szybkość, z jaką nawiązane zostaje połączenie itp. Im większa jest ta wartość tym wolniej działa program
- IP centrali – tu podany jest numer IP komputera, na którym uruchomiony jest program „centrala.exe", który potrzebny jest by symulator mógł działać w sieci.
- automatycznie lacz z centrala – tu podana jest informacja, czy symulator od razu po uruchomieniu ma się automatycznie łączyć z programem „centrala.exe" – według numeru IP podanego w sekcji „IP centrali"
- stacje brzegowe – w tej sekcji podane są nazwy stacji brzegowych, które rozpoznaje symulator. Stacje te otrzymują kolejne numery – pierwsza podana stacja ma przyznany numer 1, następna 2 itd. Numeracja ta wykorzystana jest w następnej sekcji, gdzie podane są częstotliwości, na których pracują stacje brzegowe
- czestotliwosci – jak juz wspomnialem w sekcji tej podane są częstotliwości, na których pracują stacje brzegowe. Najpierw podana jest częstotliwość, na której nadaje dana stacja, w następnej linii częstotliwość na której ta stacja odbiera, w następnej numer selcall tej stacji a poniżej numer stacji pod którym występowała ona w sekcji „stacje brzegowe".
- znamienniki stacji brzegowych – w tej sekcji podane są znamienniki stacji brzegowych w takiej kolejności, w jakiej występowały one w sekcji „stacje brzegowe"

- dostepne komendy – w tej sekcji w kolejnych wierszach podane jest, które komendy dostępne są dla poszczególnych stacji brzegowych (w kolejności takiej, w jakiej występowały w sekcji „stacje brzegowe"). Każda z linii jest ciągiem liter „t" i „n", które oznaczają, że odpowiednie komendy są lub nie są dostępne dla danej stacji brzegowej. Komendy występują tu w następującej kolejności:
  - AMV+
  - DIRTLX+
  - FREQ+
  - HELP+
  - INF+
  - MAN+
  - MED+
  - MSG+
  - NAV+
  - OBS+
  - OPR+
  - POS+
  - RTL+
  - STA+
  - STS+
  - SVC+
  - TGM+
  - TLX+
  - URG+
  - WX+
- numery abonentów – sekcja ta zawiera numery abonentów lądowych z którymi można się połączyć komendą „DIRTLX+"
- znamienniki abonentów – znamienniki abonentów z poprzedniej sekcji
- przykladowy message – ta sekcja zawiera treść wiadomości wyświetlanej po wydaniu komendy „MSG+"

- ostrzezenia nawigacyjne – w tej sekcji zawarte są ostrzeżenia nawigacyjne, które symulator wyświetla po wydaniu komendy „NAV+”
- weather – tu zapisane są informacje o pogodzie, które otrzymamy po wydaniu komendy „WX+”

6.4 Jak dopisać stację do pliku konfiguracyjnego - przykład

W poniższej części opiszę na przykładzie stacji Lyngby, jak dopisać stację brzegową do pliku konfiguracyjnego. Przede wszystkim należy pamiętać o tym, że czym innym jest dodanie stacji do pliku konfiguracyjnego a czym innym dodanie jej do książki adresowej. Książka adresowa służy jedynie wygodzie użytkownika – dzięki niej nie trzeba wybierać ręcznie częstotliwości i numeru selcall stacji – jednak to, czy stacja jest czy też nie jest wpisana do książki adresowej nie ma wpływu na to czy możemy się z nią połączyć. Nawet jeśli stacji nie ma w książce adresowej to i tak możemy się z nią połączyć jeśli jest ona wpisana do pliku konfiguracyjnego (w tym celu musimy ręcznie wpisać częstotliwości Rx i Tx oraz numer selcall). Z kolei jeśli dana stacja nie jest wpisana do pliku konfiguracyjnego to nawet jeśli dopiszemy ją do książki adresowej to i tak próba połączenia się z nią zakończy się niepowodzeniem.
W poniższym przykładzie opiszę dopisanie do pliku konfiguracyjnego stacji Lyngby. Dane dotyczące tej stacji zaczerpnąłem z „Admiralty List of Radio Signals, vol. 1(1)” z roku 1998/1999.
Zaczniemy od sekcji „stacje brzegowe”. Początkowo znajdują się w niej dwie stacje - Oostende i Scheveningen. Stację Lyngby dopiszemy jako trzecią (warto to zapamiętać - ponieważ wynika z tego, że później będziemy określać ją po prostu jako stację nr 3). Po dopisaniu stacji Lyngby do sekcji wygląda ona w ten sposób:

```
# stacje brzegowe
Oostende
Scheveningen
Lyngby
```

Teraz dopiszmy częstotliwości robocze naszej stacji do sekcji „czestotliwosci”. Pamietajmy przy tym, że stacja Lyngby ma numer kolejny „3” (numer „1” to Oostende a „2” to Scheveningen) i że dane stacji podajemy w kolejności:

- Rx (częstotliwość na której musimy odbierać - czyli częstotliwość, na której nadaje dana stacja. Jest to lewa kolumna w ALRS)

- Tx (częstotliwość na której musimy nadawać - czyli częstotliwość, na której odbiera dana stacja. Jest to prawa kolumna w ALRS)
- Numer selcall stacji - numer ten podany jest w ALRS w nawiasach kwadratowych po po nagłówku „telex”
- Numer kolejny stacji - w naszym przypadku „3”

Oczywiście dane naszej stacji możemy dopisać w dowolnym miejscu sekcji, ale dla własnej wygody najwygodniej będzie dopisać je na samym końcu sekcji. Przykładowo dla pary Rx=1613 i Tx=2148 oraz numeru selcall 0832 odbowiedni blok będzie wyglądał tak:

```
1613
2148
0832
3
```

Przejdźmy do kolejnej sekcji – znamienniki stacji brzegowej. Kolejność znamienników w tej sekcji jest taka sama, jak kolejność nazw stacji w sekcji „stacje brzegowe”. Po dopisaniu znamiennika stacji Lyngby sekcja ta wygląda jak następuje:

```
# znamienniki stacji brzegowych
0480 OSTTOR BG
2770 AUTOTX NL
0832 AUTOTX DK
```

W następnej sekcji opiszemy, które komendy są dostępne dla danej stacji. Każdej stacji odpowiada jedna linijka składająca się z 22 liter – „t” lub „n” – oznaczających, że dla tej stacji dana komenda jest dostępna lub nie. Kolejność komend w tej sekcji opisana jest w rozdziale 6.3.3 na stronie 31.
Tak uzupełniony plik konfiguracyjny gotowy jest do tego, żeby zapisać go jako „c:/telex/config.txt”; następnie należy zrestartować program aby ustawienia zostały na nowo odczytane.

Rozdział 7

Zasady działania symulatora teleksu

Program „Symulator teleksu" został napisany w języku Object Pascal, z wykorzystaniem pakietu „Delphi". Składa się on z trzech części: książki adresowej, prostego edytora tekstu i właściwego radioteleksu. Części te nie są całkowicie niezależne od siebie; jeżeli jedna z nich ma wpływ na inne odbywa się to w ten sposób, że ustawia on wartość pewnych zmiennych globalnych, które następnie są sprawdzane przez pozostałe części programu.

7.1 Plik konfiguracyjny

Istnieją pewne ustawienia programu, które muszą pozostać niezmienione przy kolejnych uruchomieniach programu; ustawienia te są zapisane w pliku „config.txt" na twardym dysku. Przy każdym uruchomieniu programu odczytywany jest plik konfiguracyjny i na jego podstawie ustawiane są wartości odpowiednich zmiennych i tablic globalnych, do których potem odwołuje się program. Zwykły użytkownik (student wykonujący ćwiczenie) nie powinien mieć możliwości zmieniania ustawień programu, dlatego też jedynym sposobem, aby zmienić ustawienia programu jest edycja pliku konfiguracyjnego, który jest zwykłym plikem tekstowym. Ponieważ może zdarzyć się, że plik konfiguracyjny zostanie przypadkowo (lub celowo) skasowany, dlatego też jeśli przy uruchomieniu programu próba otawrcia pliku konfiguracyjnego nie powiedzie się program odtworzy go ze standartowymi ustawieniami.

Plik konfiguracyjny składa się z sekcji, każda sekcja zaczyna się nagłówkiem – jedną linią rozpoczynającym się znakiem „#" po którym następuje nazwa sekcji. W następnych liniach podane są dane dotyczące tej sekcji. Odczytanie pliku konfiguracyjnego odbywa się w ten sposób, że program prze-

gląda go – linia po linii – szukając odpowiedniego nagłówka, i w ten sposób znajduje początek danej sekcji. Następnie szuka dalej, aż napotka kolejny nagłówek – i w ten sposób znaleziony jest koniec sekcji. Dalsze działanie zależy od zawartości konkretnej sekcji – w przypadku niektórych sekcji wystarczy odczytanie kolejnych linii i wpisanie ich zawartości do odpowiednich zmiennych, w przypadku sekcji zawierających bardziej złożone dane (jak na przykład spis częstotliwości na których pracują poszczególne stacje brzegowe) potrzeba podzielić sekcję na dalsze podczęści a następnie utworzyć tablicę odpowiedniej wielkości i wpisać w nią zawartość sekcji.

Strukturę pliku konfiguracyjnego oraz zawartość przykładowego pliku config.txt przedstawia schemat z załącznika.

7.2 Edytor tekstu

Prosty edytor tekstu będący częścią programu służy do edycji telegramów, tekstów do wysłania przez FEC, itp. Składa się on ze standartowego komponentu TMemo, który umożliwia wpisywanie tekstu, jego prostą edycję (kopiowanie, wycinanie, wklejanie) oraz zapis i odczyt. Obsłudze zapisu na dysk i odczytu z dysku służą polecenia „Editor – Save File" i „Editor – Open File" z głównego menu znajdującego się w formatce Form1, które wykorzystują komponenty TOpenDialog i TSaveDialog, żeby umożliwić użytkownikowi wybranie pliku do otworzenia/zapisania, a następnie wywołują odpowiednie metody komponentu TMemo. Ponieważ w edytorze tekstu powinno być możliwe wpisywanie tylko tych znaków, które należą do alfabetu ITA nr 5, więc po wciśnięciu klawisza w oknie edytora wywoływana jest procedura TForm2.Memo1KeyPress, która najpierw zamienia znak odpowiadający wciśniętemu klawiszowi (znak ten zapisany w zmiennej „key") z małej litery na dużą,a następnie sprawdza, czy tak otrzymany znak jest dostępny i jeśli nie to zastępuje go spacją.

Cała procedura wygląda w ten sposób:

```
procedure TForm2.Memo1KeyPress(Sender: TObject; var Key: Char);
begin
 key:=uppercase(key)[1];
 if (not ((key in dostepne_znaki) or (key=chr(8))))
 then key:=' ';
end;
```

, gdzie „`dostepne_znaki`" jest stałą zadeklarowaną jako:

```
const
```

```
dostepne_znaki=
               ['A','B','C','D','E','F','G','H',
                'I','J','K','L','M','N','O','P',
                'Q','R','S','T','U','V','W','X',
                'Y','Z','0','1','2','3','4','5',
                '6','7','8','9','@','#','%','(',
                ')','/','?','"','+',#10,#13];
```

Jeśli jakieś inne części programu chcą odczytać tekst z edytora sprawdzają one zmienną Form1.Memo1.text

7.3 Książka adresowa

Zawartość książki adresowej przechowywana jest w komponencie typu TMemo o nazwie Form1.KsiazkaAdresowa. Kiedy użytkownik edytuje zawartość książki adresowej zmieniana jest zawartość tego komponentu; kiedy odczytuje częstotliwości robocze danej stacji również odczytywane są one z tego komponentu. Ponadto użytkownik może zapisać zawartość książki adresowej do pliku tekstowego na dowolnym dysku w dowolnym katalogu (nie tylko w katalogu c:/telex/), może też tak zapisana książkę tekstową otworzyć – odbywa się to przez wywołanie metod SaveToFile i OpenFromFile klasy TStrings będącej elementem komonentu TMemo.
Odpowiedni fragment kodu wygląda w ten sposób:

```
procedure TForm8.Otwrz1Click(Sender: TObject);
begin
 if Form1.OpenDialog1.Execute then begin
  Form1.KsiazkaAdresowa.Lines.LoadFromFile(Form1.OpenDialog1.FileName);
 end;
 FormActivate(self);
end;

procedure TForm8.Zapisz1Click(Sender: TObject);
begin
 if Form1.SaveDialog1.Execute then begin
  Form1.KsiazkaAdresowa.Lines.SaveToFile(Form1.SaveDialog1.FileName);
 end;
end;
```

Od razu po uruchomieniu programu ksiazka adresowa powinna zawierać już kilka stacji - zapewnione jest to w ten sposób, że przy uruchomieniu programu, od razu po odczytaniu pliku konfiguracyjnego program próbuje otworzyć

plik c:/telex/adresy.txt – jeśli próba się powiedzie zawartość pliku wpisywana jest do Form1.KsiazkaAdresowa, zaś jeśli nie (to znaczy – jeśli ten plik nie stnieje) jest on tworzony i zapisywana jest w nim domyślna książka adresowa. Kiedy zawartość Form1.KsiazkaAdresowa zmienia się - na przykład od razu po uruchomieniu programu, kiedy użytkownik otworzy z dysku nową książkę adresową lub też doda do niej lub usunie z niej stację program przegląda zawartość książki adresowej, wyszukuje zawarte w niej stacje i ich listę przedstawia w komponencie Form8.Listbox1 :

```
ListBox1.Items.Clear;
for i:=0 to Form1.KsiazkaAdresowa.Lines.count-1 do begin
 if (Form1.KsiazkaAdresowa.Lines[i][1]='#') then
    ListBox1.Items.Add(copy(Form1.KsiazkaAdresowa.Lines[i],
                            2,length(Form1.KsiazkaAdresowa.Lines[i])-1));
end;
Listbox1.ItemIndex:=1;
refresh_frequencies;
```

Jak widzimy po wypisaniu wszystkich stacji w komponencie ListBox1 jedna z nich jest zaznaczana po czym wywoływana jest procedura `refresh_frequencies`. Procedura ta znajduje w książce adresowej (tj. w Form1.KsiazkaAdresowa) sekcje dotyczącą tej stacji, po czym wybiera z niej numer selcall, który wpisuje w Form8.Edit1, a następnie pary częstotliwości Rx i Tx które dopisuje do listy Form8.ComboBox1. Procedura ta wywoływana jest również wtedy, gdy użytkownik wybierze inną stację z listy Form8.ListBox1. Odpowiedni fragment kodu wygląda w ten sposób:

```
procedure TForm8.refresh_frequencies;
var
 i: integer;
 Rx_l, Tx_l, selcall_l,a:string;
begin
 //znajdujemy wybrana stacje w ksiazce
 for i:=0 to Form1.KsiazkaAdresowa.Lines.count-1 do begin
  if (Form1.KsiazkaAdresowa.Lines[i]
     ='#'+ListBox1.Items[ListBox1.ItemIndex]) then begin
   pos1:=i;
   break;
  end;
 end;
 //znajdujemy koniec sekcji tej stacji w ksiazce
 for i:=pos1+1 to Form1.KsiazkaAdresowa.Lines.count-1 do begin
```

```
  if (Form1.KsiazkaAdresowa.Lines[i][1]='#') then begin
   pos2:=i;
   break;
  end;
 end;
 selcall_l:=Form1.KsiazkaAdresowa.Lines[pos1+1];
 i:=pos1+2;
 ComboBox1.Items.Clear;
 ComboBox1.Text:='';
 while (i<pos2-1) do begin
  rx_l:=Form1.KsiazkaAdresowa.Lines[i];
  if length(rx_l)>7 then rx_l:='1000   ' else begin
   while (length(rx_l)<7) do begin
    rx_l:=rx_l+' ';
   end;
  end;
  tx_l:=Form1.KsiazkaAdresowa.Lines[i+1];
  if length(tx_l)>7 then tx_l:='1000   ' else begin
   while (length(tx_l)<7) do begin
    tx_l:=tx_l+' ';
   end;
  end;
  a:='Rx='+rx_l+'    '+'Tx='+tx_l+'  ';
  ComboBox1.Items.Add(a);
  inc(i,2);
 end;
 ComboBox1.ItemIndex:=1;
 Edit1.Text:=selcall_l;
end;
```

W ten sposób na podstawie danych zapisanych w książce adresowej tworzona jest lista stacji i odpowiadających im częstotliwości roboczych. Odwrotnej czynności należy dokonać podczas edycji książki adresowej przez użytkownika - każde dodanie stacji, zmiana nazwy czy numeru selcall, dodanie lub usunięcie pary częstotliwości roboczych powinno być związane z dopisaniem tych danych do Form1.KsiazkaAdresowa. Jest to dość proste, ponieważ edycja książki adresowej składa się z prostych czynności, takich jak dodanie pojedynczej pary częstotliwości do listy czy stworzenie nowej stacji o standartowej nazwie i numerze selcall bez przydzielonych częstotliwości roboczych. Przykładowo, usunięcie stacji odbywa się w ten sposób:

```
procedure TForm8.Usustacje1Click(Sender: TObject);
```

```
var
 i: integer;
begin
 pom.text:='';
 for i:=0 to pos1-1 do begin
  pom.lines.add(Form1.KsiazkaAdresowa.Lines[i]);
 end;
 if pos2<Form1.KsiazkaAdresowa.Lines.Count then begin
  for i:=pos2 to Form1.KsiazkaAdresowa.Lines.Count-1 do begin
   pom.lines.add(Form1.KsiazkaAdresowa.lines[i]);
  end;
 end;
 Form1.KsiazkaAdresowa.text:=pom.Text;
 FormActivate(self);
end;
```

7.4 Radioteleks

Pisząc o właściwym radioteleksie mam na myśli tę część programu, która symuluje łączność ze stacją brzegową i z innymi użytkownikami symulatora (w systemach ARQ oraz FEC). W związku z tym, że na różne działania użytkownika i inne zdarzenia program powinien reagować w różny sposób w zależności od tego, czy użytkownik aktualnie jest połączony z jakąś stacją brzegową lub też innym użytkownikiem symulatora, czy też nie, istnieje zmienna globalna Form1.Stan, która określa stan, w jakim aktualnie znajduje się teleks. Znaczenie poszczególnych wartości tej zmiennej zawiera tabela 7.1. Podstawową kwestią jest oczywiście łączenie się – ze stacją brzegową lub innym symulatorem radioteleksu. W celu połączenia się należy:

- wybrać częstotliwość
- wybrać z menu polecenie „Połącz – ARQ" lub „Połącz – FEC" albo wcisnąć odpowiedni przycisk

7.4.1 Wybór częstotliwości

Wybranie częstotliwości może odbyć się przez wybranie polecenia „Frequencies" z menu i ręczne wpisanie częstotliwości nadawczej, odbiorczej i numeru selcall lub też poprzez wejście w książkę adresową i wybranie stacji oraz pary częstotliwości roboczych z listy.

wartość zmiennej „stan”	znaczenie
0	rozłączony
1	połączony ze stacją brzegową
5	w trakcie łączenia z innym symulatorem – czeka na jego odpowiedź
6	połączony z innym symulatorem i jest stacją nadającą
7	połączony z innym symulatorem i jest stacją odbierającą
11	w trakcie wysyłania depeszy AMVER – czeka na wpisanie przez użytkownika „KKKK+”
12	w trakcie połączenia z abonentem lądowym (komenda DIRTLX+)
15	w trakcie wysyłania telegramu (komenda TGM+)

Tabela 7.1: Znaczenie zmiennej Form1.Stan

Rozważmy pierwszą możliwość: kiedy użytkownik wybiera z menu polecenie „Frequencies” wykonywany jest następujący fragment kodu:

```
procedure TForm1.Frequency1Click(Sender: TObject);
begin
 if (stan=0) then Form6.visible:=true;
end;
```

Jak widać jeśli stan=0 – to jest jeśli teleks nie jest aktualnie połączony – pokazywana jest formatka Form6. Składa się ona z trzech okienek (komponent TEdit) służących do wprowadzenia częstotliwości Rx, Tx i numeru selcall, przycisku „OK” który zatwierdza wprowadzone wartości oraz przycisku, który przenosi użytkownika do książki adresowej (rys. 6.5). Po wpisaniu częstotliwości w okienka i wciśnięciu przycisku „OK” wybrane częstotliwości i numer selcall zapamiętane są w zmiennych globalnych „Form1.Rx”, „Form1.Tx” i „Form1.Selcall”.
Wybór częstotliwości z książki adresowej był opisany w rozdziale 7.3 na stronie 38

7.4.2 Łączenie

Po wybraniu częstotliwości można wcisnąć przycisk „ARQ” lub „FEC” albo wybrać odpowiednie polecenie z menu „Connect”.

ARQ

Połączenie ARQ możliwe jest zarówno ze stacją brzegową, jak i z użytkownikami symulatora na innych komputerach. Użytkownik łącząc się nie precyzuje, z jakim rodzajem stacji będzie się łączył (tzn. ze stacją brzegową czy z innym symulatorem), podaje tylko numer selcall i częstotliwości Rx i Tx. Dlatego po wydaniu polecenia „ARQ” program napierw przegląda tablicę Form1.Czestotliwosci, w ktorej zapisane są częstotliwości i numery selcall wszystkich dostępnych stacji brzegowych. Jeśli okaże się, że podane częstotliwości i numery selcall pasują do którejś ze stacji brzegowych to (niezależnie od tego, czy stacja ta jest wpisana do książki adresowej czy też nie) zostaje nawiązane połączenie z tą stacją. Jeśli nie – program próbuje połączyć się z innym symulatorem, o numerze selcall i częstotliwościach Rx i Tx zgodnych z podanymi przez użytkownika.

Połączenie ARQ ze stacją brzegową Po odnalezieniu stacji brzegowej w Form1.Czestotliwosci wartość zmienna Form1.Stan przyjmuje wartość 1 (patrz tablica 7.1 na stronie 42), a w odpowienich zmiennych zapamiętywane

są dane dotyczące stacji brzegowej, z którą jesteśmy połączeni (szczegółowo jest to opisane w tabeli 7.2 na stronie 45). Odpowiedni fragment programu wygląda w ten sposób:

```
if PasujeStacjaBrzegowa then begin
 connected:=true;
 j:=StrToInt(czestotliwosci[l+3]);
 ZnamiennikStacjiBrzegowej:=ZnamiennikiStacjiBrzegowych[j-1];
 DostepneAktualnieKomendy:=DostepneKomendy[j-1];
 ZnamiennikRozmowcy:=ZnamiennikStacjiBrzegowej;
 PolaczonyZeStacjaNr:=j;
 //zaczynamy wypisywanie na ekran
 delay(random(30));
 // napisz ze jestesmy polaczeni
 form3.StatusBar1.Panels[0].text:='Connected';
 SpeedButton1.Enabled:=false;
 SpeedButton2.Enabled:=false;
 SpeedButton5.Enabled:=true;
 SpeedButton6.Enabled:=true;
 SpeedButton7.Enabled:=false;
 form3.StatusBar1.refresh;
 {wypisuje answerback stacji}
 pisz(ZnamiennikStacjiBrzegowej+enter);
 form1.refresh;
 delay(4);
 {wypisuje nasz answerback}
 pisz(MojZnamiennik+enter);
 form1.refresh;
 {znowu czeka}
 delay(4);
 {no i na koniec pisze "GA+"}
 pisz('GA+? '+enter);
 // napisz ze gadamy
 form3.StatusBar1.Panels[4].text:='Speaking';
 form3.StatusBar1.refresh;
 // stan = 1 czyli "polaczony ze stacja brzegowa, normalny tryb"
 stan:=1;
end
```

Połączenie ARQ z innym komputerem Cała łączność między różnymi komputerami odbywa się za pośrednictwem uruchomionego na jednym

zmienna	znaczenie
Form1.Connected	teleks jest w stanie „połączony”
Form1.ZnamiennikStacjiBrzegowej	znamiennik stacji brzegowej, z którą jesteśmy aktualnie połączeni
Form1.DostepneAktualnieKomendy	zmienna określająca, które komendy są dostępne przy połączeniu z aktualną stacją brzegową (patrz rozdział 6.3.3 na stronie 31)
Form1.ZnamiennikRozmowcy	znamiennik naszego rozmowcy – początkowo jest to znamiennik stacji brzegowej, z którym jesteśmy połączeni; kiedy użytkownik połączy się z abonentem lądowym (przy użyciu komendy DIRTLX+) będzie tu zapamiętany jego znamiennik
Form1.PolaczonyZeStacjaNr	numer kolejny stacji brzegowej, z którą jesteśmy połączeni
Form1.Stan	stan teleksu – wartość „1” oznacza „połączony ze stacją brzegową”

Tabela 7.2: Znaczenie poszczególnych zmiennych przy połączeniu ARQ ze stacją brzegową

z komputerów programu „centrala.exe” (patrz rysunek 6.6 na stronie 30). Służy do tego komponent Form1.ClientSocket1 typu TClientSocket:

```
object ClientSocket1: TClientSocket
  Active = False
  ClientType = ctNonBlocking
  Port = 1024
  OnLookup = ClientSocket1Lookup
  OnConnecting = ClientSocket1Connecting
  OnConnect = ClientSocket1Connect
  OnRead = ClientSocket1Read
  Left = 152
  Top = 120
end
```

Kiedy program łączy się z centralą – albo od razu przy uruchomieniu programu (jeśli tak jest ustawione w pliku konfiguracyjnym, patrz rozdział 6.3.3 na stronie 31), albo na żądanie użytkownika, poprzez wybranie polecenia „Łącz z centralą” z menu „System” – do zmiennej Form1.ClientSocket1.Address wpisujemy numer IP centrali (przechowywany w zmiennej Form1.IPCentrali) a zmiennej Form1.ClientSocket1.Active nadajemy wartość „true”. Odpowiedni fragment programu wygląda w ten sposób:

```
procedure TForm1.laczzcentrala1Click(Sender: TObject);
begin
  form1.clientsocket1.Address :=form1.IPCentrali;
  form1.clientsocket1.active:=true;
  form1.PolaczonyZCentrala:=true;
end;
```

Wysyłanie tekstu do centrali odbywa się przy użyciu metody SendText komponentu TClientSocket. Po odebraniu tekstu „centrala” rozyła go bez żadnych zmian do wszystkich podłączonych do niej symulatorów. Po odebraniu tego tekstu (komunikatu) przez TClientSocket1 wywoływane jest zdarzenie OnRead, związane z procedurą ClientSocket1OnRead. Procedura sprawdza do kogo zaadresowany był rozesłany komunikat i co zawierał (przesyłany tekst, wywołanie itp) i reaguje w odpowiedni sposób.
Struktura komunikatu przedstawiona jest na rysunku 7.1 na stronie 49. Jak widać komunikat składa się z dwu głównych części: nagłówka i treści. Nagłówek informuje do kogo skierowany jest ten komunikat – zawiera on częstotliwość, na której nadano dany komunikat (zawsze siedem cyfr) oraz numer selcall stacji, do której jest skierowany (również siedem cyfr). Treść może

zawierać tylko przesyłany tekst, może zawierać również komendy które określają rodzaj komunikatu - np. wywołanie, odpowiedź na wywołanie, FEC czy rozłączenie. Komendy są ciągami czterech znaków, ich spis i znaczenie przedstawia tabela 7.3 na stronie 50
Jeśli symulator o numerze selcall *0001* odbierający na częstotliwości*1000* chce się połączyć z symulatorem o numerze *00002* odbierającym na częstotliwości *1500* powinien on mieć przede wszystkim częstotliwość Tx (częstotliwość nadawania - zmienna Form1.Tx) ustawioną na *1500* oraz numer selcall (zmienna Form1.Selcall) ustawiony na *00002*. Po wywołaniu polecenia ARQ z menu „Connect" oraz sprawdzeniu, że podanym częstotliwościom oraz numerowi selcall nie odpowiada żadna stacja brzegowa program wysyła do centrali komunikat o treści:

`1500␣␣␣00002␣␣@hop!00001␣PROGRAM␣PC`

Jak widać pierwsze siedem znaków to częstotliwość, na jakiej nadane zostało wywołanie (ponieważ częstotliwość ta jest liczbą czterocyfrową więc dopełniono ją trzema spacjami), następne siedem znaków to numer selcall wywoływanej stacji - również dopełniony spacjami. Od piętnastego znaku zaczyna się właściwa treść komunikatu. Jej pierwszy znak to „`@`", co oznacza, że komunikat ten zawiera jakąś komendę (zwykłe komunikaty, służące do przesyłania tekstu między stacjami które nawiązały już łączność nie zawierają żadnej komendy). Następne cztery znaki to komenda która oznacza, że ten komunikat jest wywołaniem stacji o podanym numerze selcall. Końcówka komunikatu zawiera znamiennik stacji wywołującej.
Po odebraniu komunikatu centrala rozsyła go do wszystkich symulatorów. Każdy symulator otrzymawszy komunikat sprawdza, czy był on skierowany do niego (sprawdzając pierwszych 14 znaków komunikatu, które zawierają częstotliwość i numer selcall). Jeśli tak, i jeśli wywoływana stacja jest w stanie „0" (patrz tabela 7.1 na stronie 42) to wtedy zmienia swój stan na „6" i odpowiada komunikatem o treści:

`1000␣␣␣00001␣␣@odp!00002␣PROGRAM␣PC`

Podobnie jak w poprzednim komunikacie widzimy, że pierwszych 14 znaków zawiera częstotliwość i numer selcall. Następnie widzimy komendę „odp!", oznaczającą odpowiedź na wywołanie oraz znamiennik stacji wywoływanej – „00002 PROGRAM PC".
W wyniku takiej wymiany komunikatów stacje są połączone, jedna z nich jest stacją nadającą (stan 6) a druga odbierającą (stan 7).

FEC

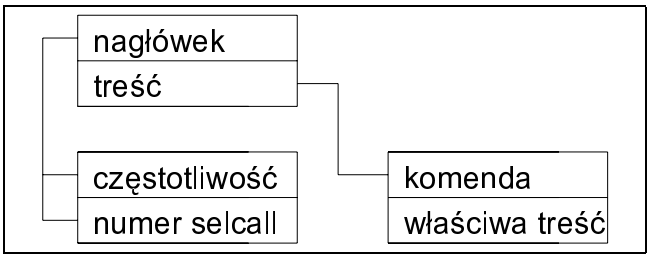
Wysłanie tekstu do innych komputerów jako FEC odbywa się przez wciśnięcie przycisku „FEC" lub wybranie polecenia „FEC" z menu „Connect". Procedura TForm1Fec1Click wysyła do centrali komunikat składający się z nagłówka – zawierającego częstotliwość i „0000000" zamiast numeru selcall – oraz treści, zaczynającej się komendą „fec!" po której następuje zawartość edytora – czyli Form2.Memo1.Text.
Każda stacja, która znajduje się w stanie „0" po odebraniu takiego komunikatu sprawdza, czy komunikat ten był nadany na jej częstotliwości nasłuchu i jeśli tak to wypisuje przesłany tekst na ekranie, po czym w dalszym ciągu znajduje się w stanie „0" - gotowa do nawiązania połączenia lub odebrania kolejnej wiadomości FEC.

7.4.3 Wydawanie komend stacji brzegowej

Wydawanie komend stacji brzegowej możliwe jest jeżeli teleks znajduje się w stanie „1". Każdy znak wpisany w głównym oknie teleksu dopisywany jest do bufora (Form1.Bufor). Jeżeli teleks znajduje się w stanie „1" a wciśniętym klawiszem jest „+" wywoływana jest procedura „`Form1.Over_1`". W procedurze tej program sprawdza, czy bufor zawiera którąś ze znanych komend a jeśli tak, to wywoływana jest procedura obsługująca tę komendę; następnie bufor jest czyszczony. Niektóre komendy (na przykład TGM+) wymagają, żeby po ich wydaniu użytkownik wpisał jakiś tekst (na przykład treść telegramu), po czym wpisał „KKKK+" co zakończy obsługę tej komendy. Dlatego też wydanie tej komendy powoduje, że stan teleksu jest zmieniony na jakąś specyficzną wartość (np. w przypadku komendy TGM+ jest to stan 15). Kiedy teraz teleks znajduje się w tym stanie i użytkownik wpisze „KKKK+" to wywoływana jest procedura `TForm1.Over_15`, która wypisuje na ekran potrzebne informacje, zmienia stan teleksu z powrotem na „1" i kończy w ten sposób obsługę komendy „TGM+".

7.4.4 Wymiana informacji między dwoma symulatorami w systemie ARQ

Wymiana informacji między dwoma symulatorami odbywa się w ten sposób, że zawsze jedna stacja ma status stacji nadającej, a druga odbierającej. Kiedy użytkownik stacji nadającej wpisuje jakiś tekst jest on przechowywany w buforze (buforem tym jest zmienna tekstowa Form1.Bufor). Po wpisaniu „+?" tekst ten jest przesłany do stacji odbierającej, bufor jest czyszczony a stacja zmienia swój stan z „6" na „7"– czyli ze stacji nadającej na odbiera-



Rysunek 7.1: Struktura komunikatu przesyłanego między symulatorem a centralą

jącą. Komunikat z przesłanym tekstem składa się z nagłówka, zawierającego częstotliwość i numer selcall, oraz właściwej treści komunikatu, która zawiera tylko przesyłany tekst (nie zawiera żadnych komend).
Po wybraniu przez użytkownika z menu polecenia „Disconnect" teleks przechodzi w stan „0" (rozłączony) i wypisuje na ekran informację o rozłączeniu.

7.5 Obsługa drukarki

Użytkownik programu ma możliwość włączenia bieżącego drukowania wszystkiego, co jest wypisywane na ekran. Odbywa się to przez wciśnięcie przycisku „Włącz/ wyłącz drukowanie". Kiedy przycisk ten jest wciśnięty wszystko, co jest wypisywane na ekran jest również wysyłane do LPT1 - niestety, opcja ta działa praktycznie tylko dla drukarek igłowych, ponieważ tylko one dają możliwość drukowania „wiersz po wierszu".

komenda	znaczenie
„hop!”	Ta komenda wysyłana jest przez stację wywołującą do stacji wywoływanej. Oznacza ona: „Chcę nawiązać z tobą łączność. Jeśli odebrałeś to wywołanie to odpowiedz”
„odp!”	Tę komendę wysyła stacja wywoływana jako odpowiedź do stacji, która ją wywoływała. Oznacza ona: „odebrałem twoje wywołanie, od tej chwili łączność możemy uznać za nawiązaną”
„fec!”	Ta komenda oznacza, że przesyłany komunikat zawiera wiadomość przesyłaną jako FEC. Po odebraniu takiego komunikatu stacja, która ją odebrała wypisuje ją na ekran, chyba że jest już połączona z inną stacją.
„BRK!”	– „rozłączmy się”. Ta komenda wysłana jest w celu rozłączenia się.

Tabela 7.3: Znaczenie komend zawartych w komunikatach przesyłanych między symulatorem a centralą

Rozdział 8

Skorowidz

Skorowidz

Spis literatury

[1] Stefan Hahn, *Podstawy radiokomunikacji*, Wydawnictwa Komunikacji i Łączności, 1964

[2] prof. dr inż. Witold Nowicki, *Podstawy teletransmisji*, Wydawnictwa Komunikacji i Łączności, 1974

[3] praca zbiorowa pod red. Włodzimierza Trusza, *Teleelektronika*, Wydawnictwa Komunikacji i Łączności, 1974

[4] J. Machlowski, *Zarys telekomunikacji kolejowej*, Wydawnictwa Komunikacji i Łączności, 1983

[5] K. Pogoda, *Technika i technologia telekomunikacji*, Wydawnictwo Naukowe Uniwersytetu Szczecińskiego, 1996

[6] praca zbiorowa pod redakcją Jerzego Czajkowskiego, *System GMDSS, regulaminy, procedura i obsługa*, PWP Skryba, 2000

[7] *Admiralty List of Radio Signals, vol. 1(1)*, Hydrographer of the Navy, 1998

[8] Barbara Katarzyńska, *Ship's Correspondence*, Fundacja Rozwoju Wyższej Szkoły Morskiej w Gdyni

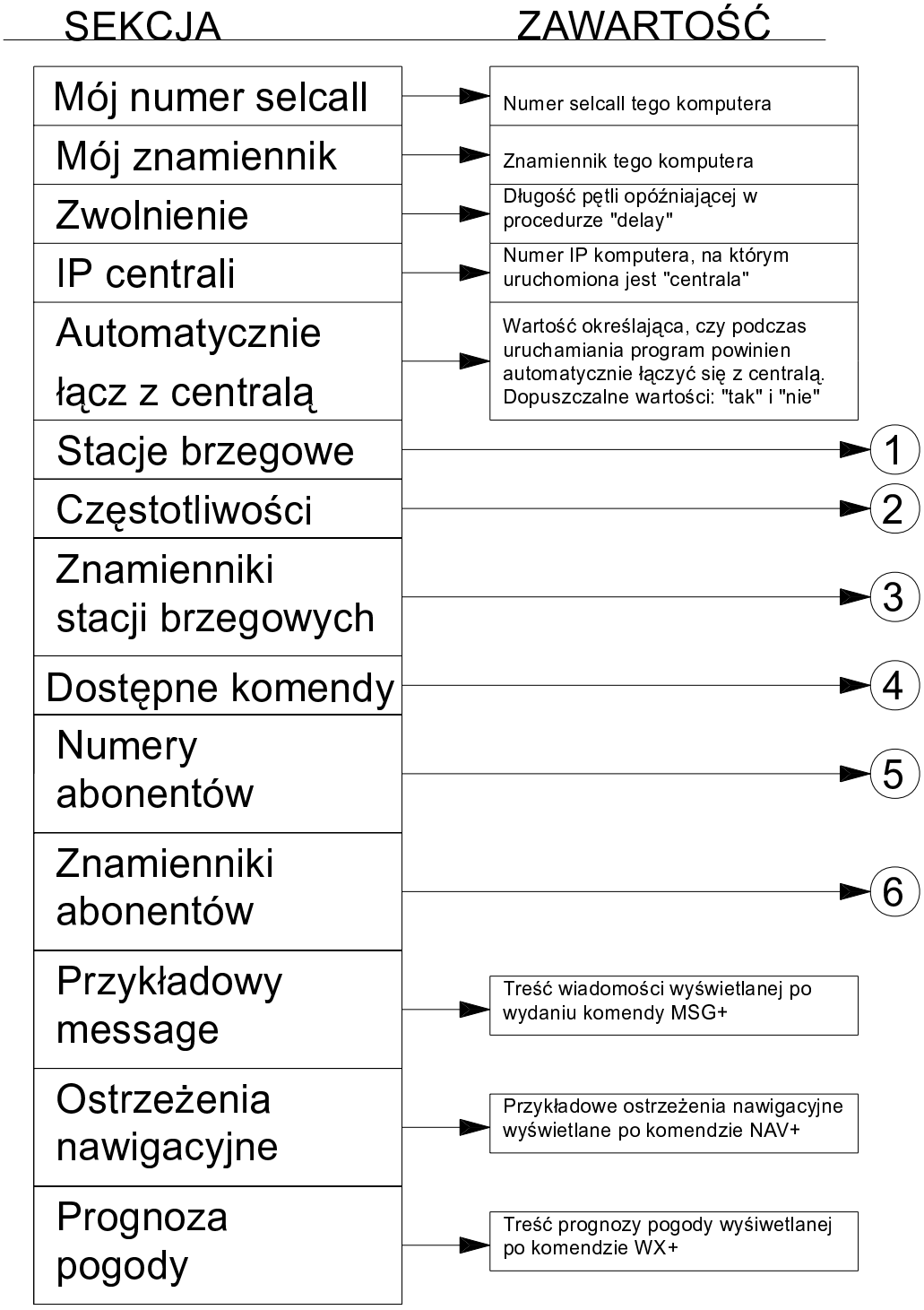
Rozdział 9

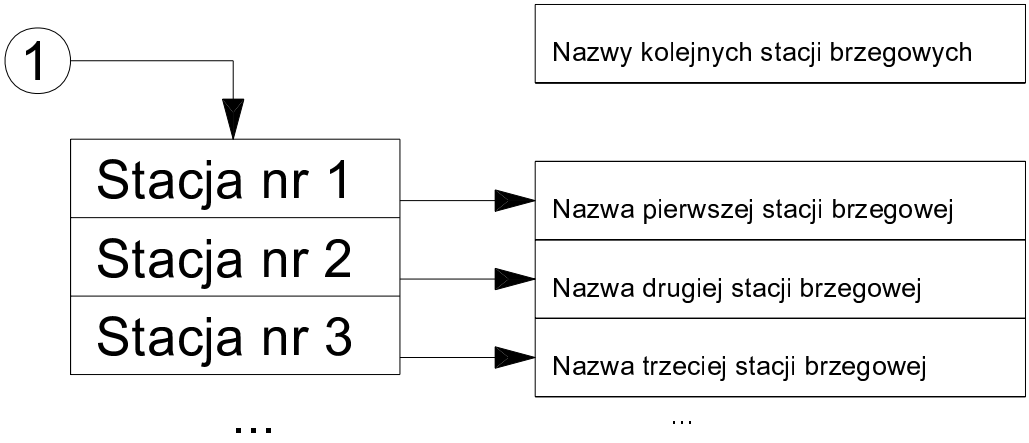
Załączniki

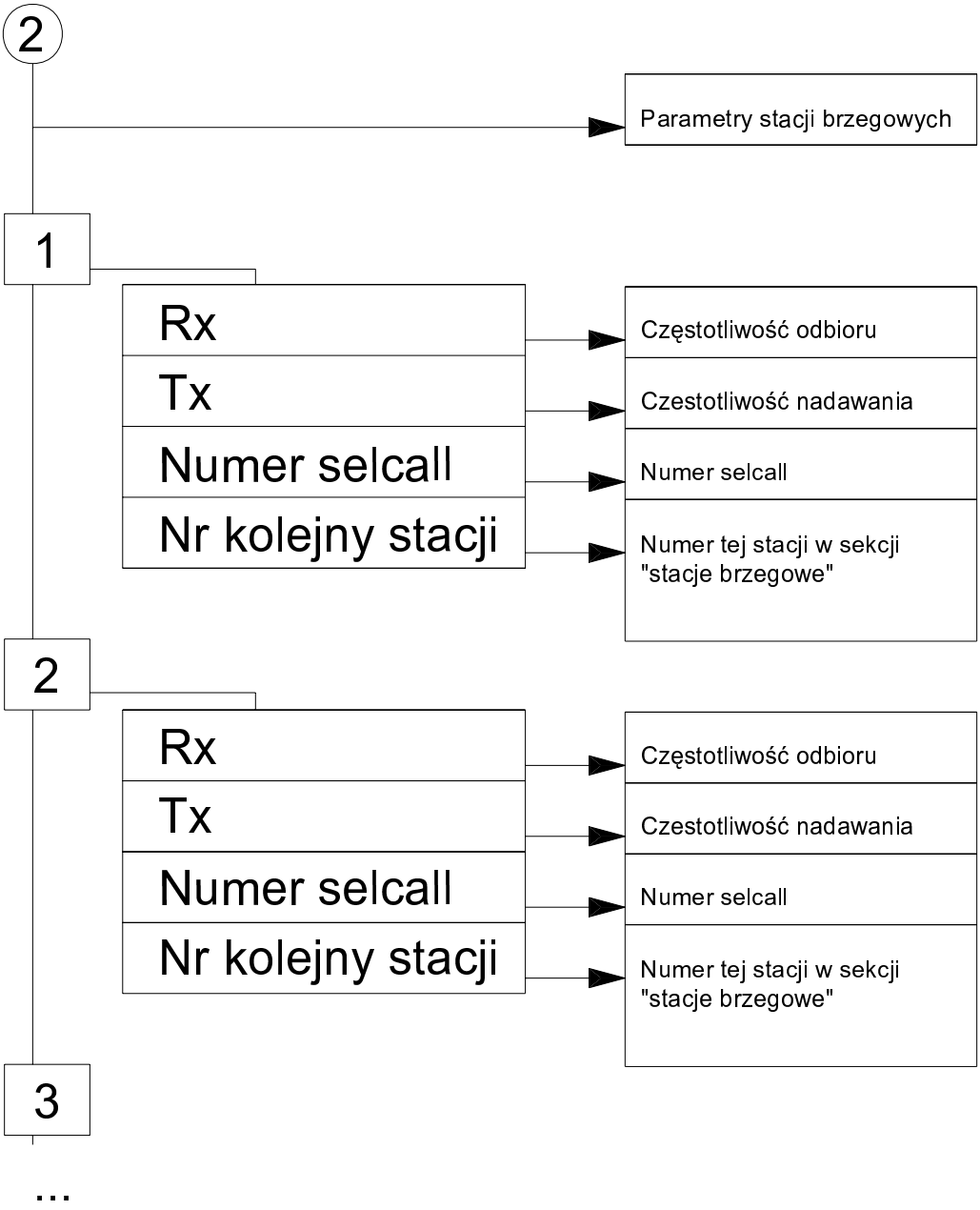
9.1 Instrukcja do ćwiczenia „Łączność radioteleksowa w systemie ARQ i FEC”

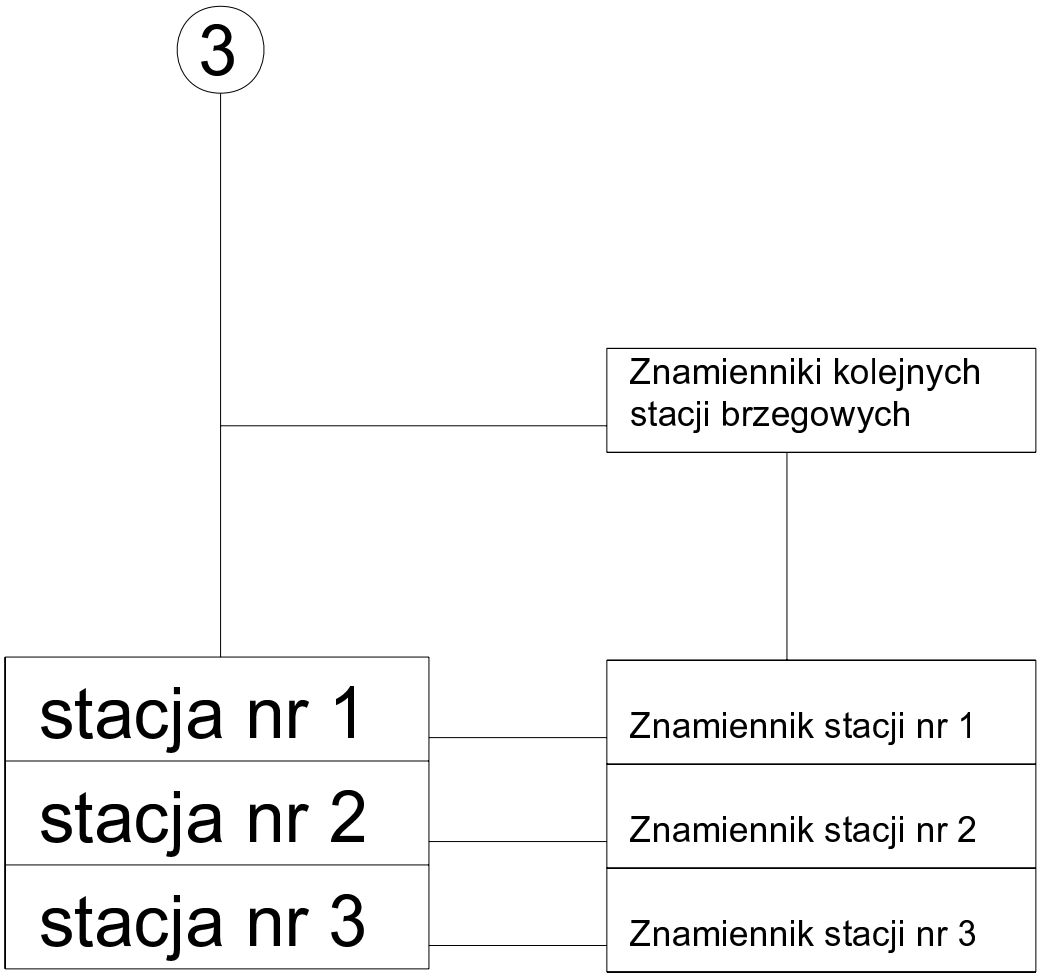
9.2 Algorytm programu

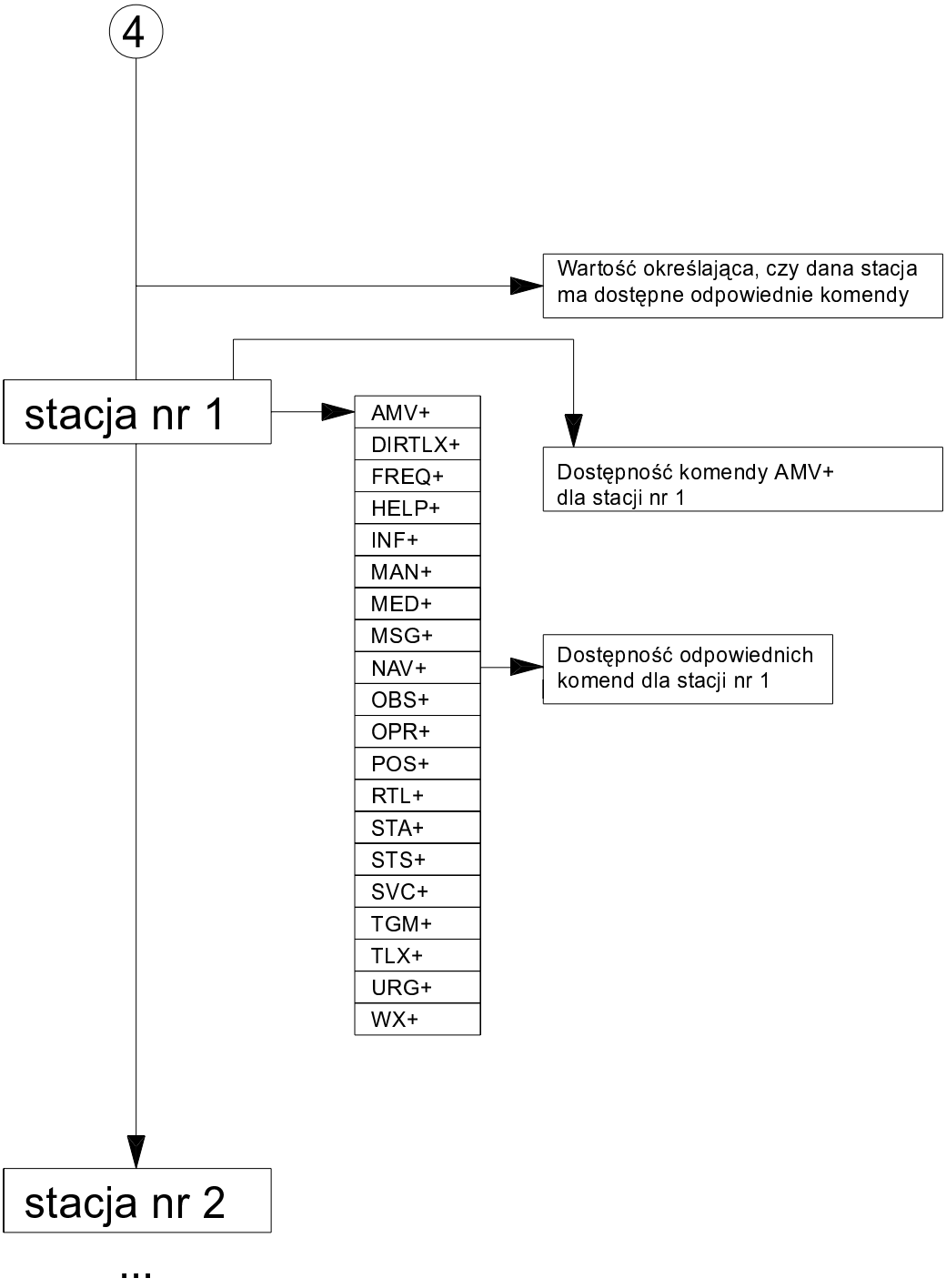
9.3 Struktura pliku konfiguracyjnego

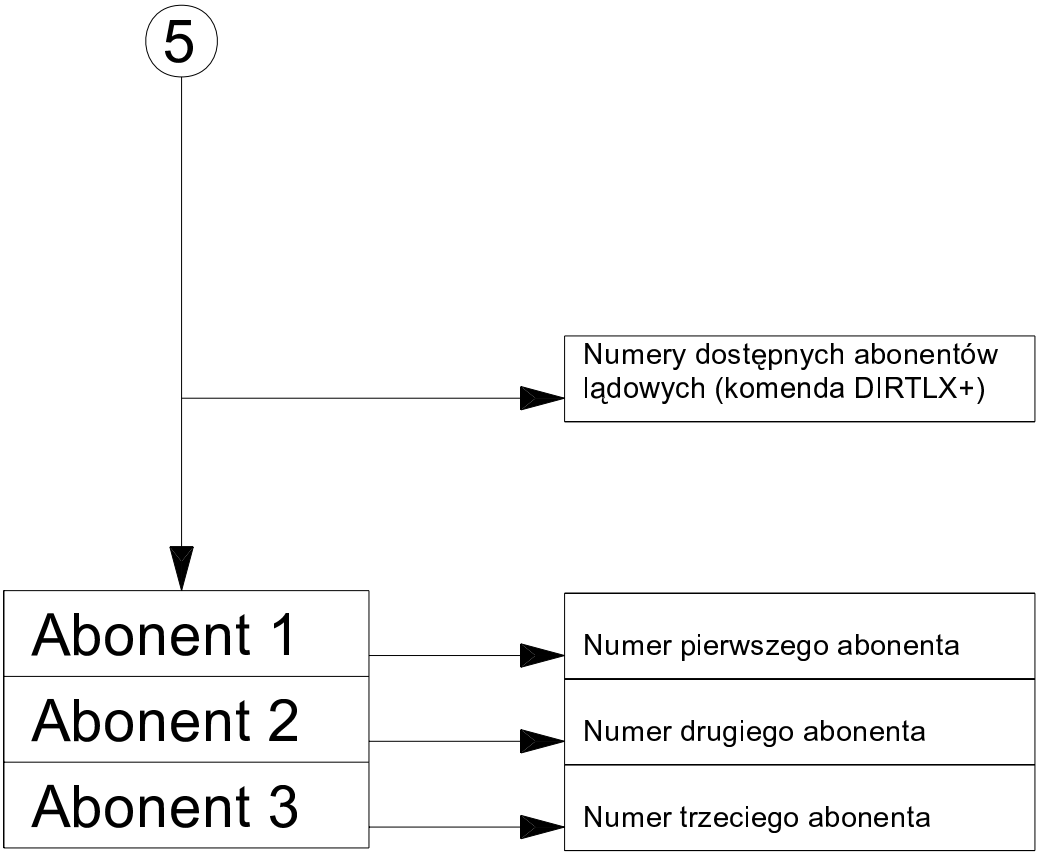

SEKCJA
ZAWARTOŚĆ
Mój numer selcall
Numer selcall tego komputera
Mój znamiennik
Znamiennik tego komputera
Zwolnienie
Długość pętli opóźniającej w procedurze "delay"
IP centrali
Numer IP komputera, na którym uruchomiona jest "centrala"
Automatycznie łącz z centralą
Wartość określająca, czy podczas uruchamiania program powinien automatycznie łączyć się z centralą. Dopuszczalne wartości: "tak" i "nie"
Stacje brzegowe
1
Częstotliwości
2
Znamienniki stacji brzegowych
3
Dostępne komendy
4
Numery abonentów
5
Znamienniki abonentów
6
Przykładowy message
Treść wiadomości wyświetlanej po wydaniu komendy MSG+
Ostrzeżenia nawigacyjne
Przykładowe ostrzeżenia nawigacyjne wyświetlane po komendzie NAV+
Prognoza pogody
Treść prognozy pogody wyśiwetlanej po komendzie WX+

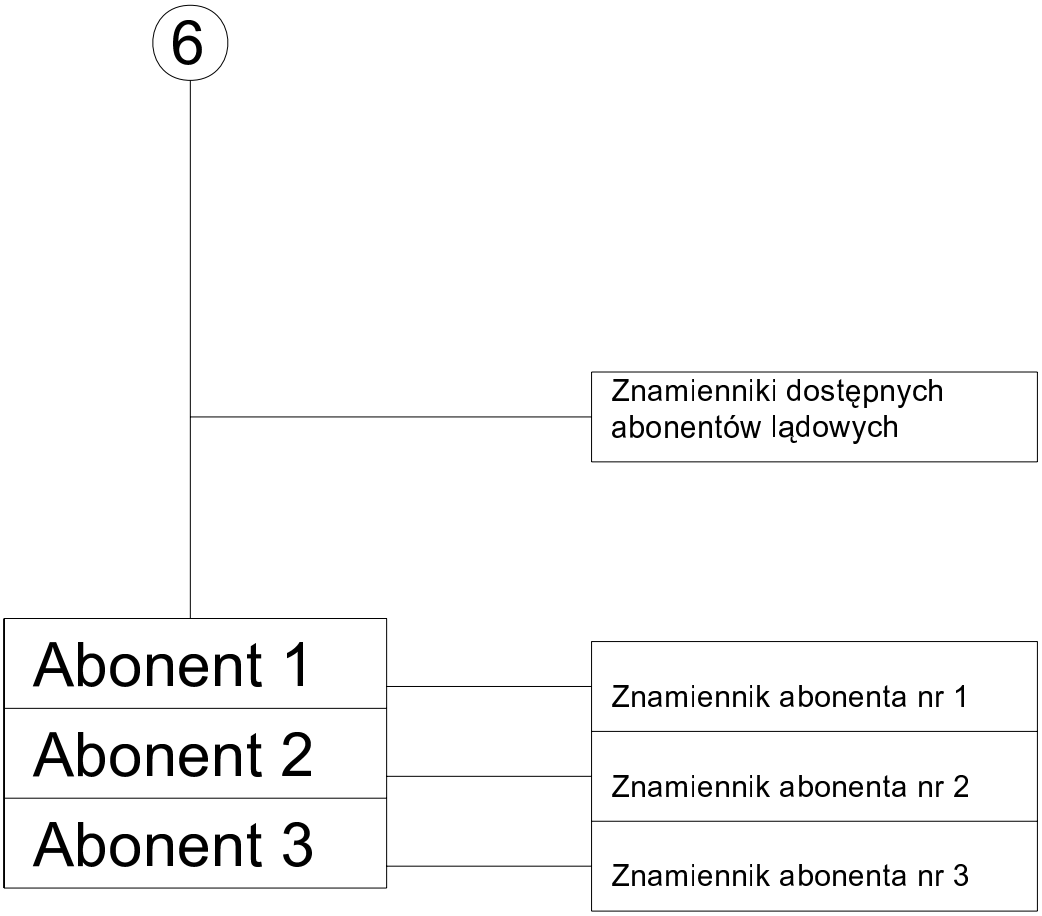

1
Nazwy kolejnych stacji brzegowych
Stacja nr 1
Stacja nr 2
Stacja nr 3
...
Nazwa pierwszej stacji brzegowej
Nazwa drugiej stacji brzegowej
Nazwa trzeciej stacji brzegowej
...


2
Parametry stacji brzegowych
1
Rx
Tx
Numer selcall
Nr kolejny stacji
Częstotliwość odbioru
Czestotliwość nadawania
Numer selcall
Numer tej stacji w sekcji "stacje brzegowe"
2
Rx
Tx
Numer selcall
Nr kolejny stacji
Częstotliwość odbioru
Czestotliwość nadawania
Numer selcall
Numer tej stacji w sekcji "stacje brzegowe"
3
...


3
Znamienniki kolejnych stacji brzegowych
stacja nr 1
stacja nr 2
stacja nr 3
Znamiennik stacji nr 1
Znamiennik stacji nr 2
Znamiennik stacji nr 3


4
Wartość określająca, czy dana stacja ma dostępne odpowiednie komendy
stacja nr 1
AMV+
DIRTLX+
FREQ+
HELP+
INF+
MAN+
MED+
MSG+
NAV+
OBS+
OPR+
POS+
RTL+
STA+
STS+
SVC+
TGM+
TLX+
URG+
WX+
Dostępność komendy AMV+ dla stacji nr 1
Dostępność odpowiednich komend dla stacji nr 1
stacja nr 2
...


5
Numery dostępnych abonentów lądowych (komenda DIRTLX+)
Abonent 1
Abonent 2
Abonent 3
Numer pierwszego abonenta
Numer drugiego abonenta
Numer trzeciego abonenta


6
Znamienniki dostępnych abonentów lądowych
Abonent 1
Abonent 2
Abonent 3
Znamiennik abonenta nr 1
Znamiennik abonenta nr 2
Znamiennik abonenta nr 3

9.4 Przykładowy plik konfiguracyjny

```
# moj numer selcall
00001
# moj znamiennik
00001 PROGRAM PC
# zwolnienie
300
# IP centrali
0.0.0.0.
# automatycznie lacz z centrala
nie
# stacje brzegowe
Oostende
Scheveningen
# czestotliwosci
2815.5
1971.5
0480
1
4218
4180.5
0480
1
6322
6271
0480
1
8435.5
8395.5
0480
1
12639.5
12537.5
0480
1
16883
16765
0480
1
19698
```

```
18887.5
0480
1
22443
22351
0480
1
1619.5
2154.5
2770
2
4217
4179.5
2770
2
8428.5
8388.5
2770
2
12585
12482.5
2770
2
12596.5
12494
2770
2
16826.5
16703.5
2770
2
16839
16716
2770
2
22380
22288
2770
2
1619.5
2154.5
```

```
2771
2
4217
4179.5
2771
2
8428.5
8388.5
2771
2
12585
12482.5
2771
2
12596.5
12494
2771
2
16826.5
16703.5
2771
2
16839
16716
2771
2
22380
22288
2771
2
# znamienniki stacji brzegowych
0480 OSTTOR BG
2770 AUTOTX NL
# dostepne komendy
nttttttttttttttttttt
ttttnttttttttnttttttt
# numery abonentow
0636511340
0630123456
0631122330
# znamienniki abonentow
```

```
6511340 TX PL
0123456 TX PL
1122330 TX PL
# przykladowy message
MESSAGE NR SO 0001
MESSAGE SVC       REF:A064Q 2000 05 11 19:54
QRJ NOREF: A064Q FM: 11 MAY 2000 TO: PIOTR SOBOLEWSKI
USE PHONE CH: 294/810/1227 OR VHF/26/27/25=SWIN-ROAD,
24/28=SZCZECIN-PORT   OR CALL VIA RTLX/SPB OPR+
# ostrzezenia nawigacyjne
23022000     2130 UTC
NOORD HINDER TRAFFIC SEPERATION SCHEME
SURVEY OPERATIONS BAY HNLNS TYDEMAN IN
NORTHEAST BOUND LANE AND PRECAUTIONARY
AREA BETWEEN TWIN BUOY AND NOORDHINDER LIGHTVESSEL

01032000     1345 UTC
WESTHINDER FLUSHING SCHEUR ROUTE 51 25 09.3, 03 17 59.5
TEMPORARILY ESTABLISHED IN THE
ABOVE MENTIONED POSITION A HYDROGRAPHICAL INSTRUMENT
ABOUT 50 METRES OF THE
METALLIC STRUCTURE MOW4. A WIDE BERTH IS REQUIRED.
# weather

22.03.2000     1245UTC
WARNINGS - WEAKENING GALE 998 MB 51 N 164W
MOVING SLOWLY
EAST 10 KT. WIND 35 KT SEAS 20 FT WITHIN 300 M
SOUTH SEMICIRCLE AND
200 M ELSEWHERE

THE GENERAL SYNOPSIS AT MIDDAY
LOW NEAR SOUTHWESTERN
FINISTEIRE 1003 EXPECTED CENTRAL ENGLAND
992
BY MIDDAY TOMORROW STOP

THE GENERAL SYNOPSIS AT MIDDAY - LOW
CENTRAL
ENGLAND 996 EXPECTED BALTIC 975 BY MIDDAY
TOMORROW
```

LOW HEBRIDES 989 SLOW MOVING AND FILLING
998 BY SAME TIME.
#

9.5 Przykładowy plik książki adresowej

```
#Oostende
0480
2815.5
1971.5
4218
4180.5
6322
6271
8435.5
8395.5
12639.5
12537.5
16883
16765
19698
18887.5
22443
22351
#Scheveningen
2770
1619.5
2154.5
4217
4179.5
8428.5
8388.5
12585
12482.5
12596.5
12494
16826.5
16703.5
16839
16716
22380
22288
1619.5
2154.5
4217
```

```
4179.5
8428.5
8388.5
12585
12482.5
12596.5
12494
16826.5
16703.5
16839
16716
22380
22288
#}
```

9.6 Kod źródłowy programu

9.6.1 Unit1.pas

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Menus, Buttons,
  ToolWin, ComCtrls, ExtCtrls, ScktComp;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    Call1: TMenuItem;
    Frequency1: TMenuItem;
    Commands1: TMenuItem;
    Applications1: TMenuItem;
    System1: TMenuItem;
    Help1: TMenuItem;
    Radiotelex1: TMenuItem;
    Editor1: TMenuItem;
    Telex1: TMenuItem;
    newtelex1: TMenuItem;
    opentelex1: TMenuItem;
    savetelex1: TMenuItem;
    SaveDialog1: TSaveDialog;
    OpenDialog1: TOpenDialog;
    savetelexas1: TMenuItem;
    ARQ1: TMenuItem;
    FEC1: TMenuItem;
    ControlBar1: TControlBar;
    Panel1: TPanel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    ClientSocket1: TClientSocket;
    laczzcentrala1: TMenuItem;
    TEST1: TMenuItem;
    socketonread1: TMenuItem;
    stan1: TMenuItem;
    Timer1: TTimer;
```

```
Whoareyou1: TMenuItem;
Thisis1: TMenuItem;
Sendfile1: TMenuItem;
Sendtext1: TMenuItem;
Memo1: TMemo;
Panel2: TPanel;
SpeedButton3: TSpeedButton;
SpeedButton4: TSpeedButton;
Panel3: TPanel;
SpeedButton5: TSpeedButton;
SpeedButton6: TSpeedButton;
selcall1: TMenuItem;
KsiazkaAdresowa: TMemo;
AddressBook1: TMenuItem;
Panel4: TPanel;
SpeedButton7: TSpeedButton;
stan2: TMenuItem;
Over1: TMenuItem;
Panel5: TPanel;
SpeedButton8: TSpeedButton;
SpeedButton9: TSpeedButton;
SpeedButton10: TSpeedButton;
procedure Editor1Click(Sender: TObject);
procedure Radiotelex1Click(Sender: TObject);
procedure newtelex1Click(Sender: TObject);
procedure savetelex1Click(Sender: TObject);
procedure savetelexas1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Frequency1Click(Sender: TObject);
procedure ARQ1Click(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure ClientSocket1Read(Sender: TObject; Socket: TCustomWinSocket);
procedure ClientSocket1Connect(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ClientSocket1Connecting(Sender: TObject;
  Socket: TCustomWinSocket);
procedure ClientSocket1Lookup(Sender: TObject;
  Socket: TCustomWinSocket);
procedure laczzcentrala1Click(Sender: TObject);
procedure socketonread1Click(Sender: TObject);
procedure stan1Click(Sender: TObject);
```

```
  procedure FEC1Click(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
  procedure Whoareyou1Click(Sender: TObject);
  procedure Thisis1Click(Sender: TObject);
  procedure Sendfile1Click(Sender: TObject);
  procedure Sendtext1Click(Sender: TObject);
  procedure SpeedButton4Click(Sender: TObject);
  procedure SpeedButton3Click(Sender: TObject);
  procedure SpeedButton5Click(Sender: TObject);
  procedure SpeedButton6Click(Sender: TObject);
  procedure selcall1Click(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);
  procedure AddressBook1Click(Sender: TObject);
  procedure SpeedButton7Click(Sender: TObject);
  procedure stan2Click(Sender: TObject);
  procedure Over1Click(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
  procedure SpeedButton9Click(Sender: TObject);
  procedure SpeedButton10Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
  Test_ReceiveText: string;
  TEST: boolean;

  enter: string;
  connected: boolean;
  rx,tx: real;
  selcall:integer;
  sciezka: string;
  MojNumerSelcall: integer;
  MojZnamiennik: string;
  Zwolnienie: integer;
  StacjeBrzegowe: array of string;
  Czestotliwosci: array of string;
  ZnamiennikiStacjiBrzegowych: array of string;
  DostepneKomendy: array of string;
  NumeryAbonentow: array of integer;
  ZnamiennikiAbonentow: array of string;
  stan: byte;
```

```
{ stany:
        0 - rozlaczony
        1 - polaczony ze stacja brzegowa, tryb normalny
        5 - wlasnie laczy z innym statkiem i czeka na jego odpowiedz
        6 - polaczony ze statkiem - ja slucham
        7 - polaczony ze statkiem - ja mowie
        11 - wysylanie depeszy amver
        12 - polaczenie z abonentem z ladu
        13 - czyli po prostu freq
        14 - wysylanie obs (polecenie obs+)
        15 - wysylanie TGM
}
ZnamiennikStacjiBrzegowej: string;
DostepneAktualnieKomendy: string;
ZnamiennikAbonenta: string;
ZnamiennikRozmowcy: string;
IPcentrali: string;
LaczZCentrala: boolean;
PolaczonyZCentrala: boolean;
PolaczonyZeStacjaNr: integer;
polecenia: array of string;
OpisPolecen: array of string;
bufor: string;
JestMessage: boolean;
PrzykladowyMessage: string;
OstrzezeniaNawigacyjne: string;
weather: string;
procedure delay(czas: integer);
procedure pisz(co: string);
procedure pisz_czas;
procedure over_1;
procedure over_7;
procedure over_11;
procedure over_12;
procedure over_13;
procedure over_14;
procedure over_15;
procedure kHelp;
procedure kBrk;
procedure kAmv;
procedure kDirtlx(komenda: string);
```

```
    procedure kTlx(komenda: string);
    procedure kFreq;
    procedure kMed;
    procedure kMsg;
    procedure kNav;
    procedure kObs;
    procedure kOpr;
    procedure kTgm;
    procedure kWx;
    procedure kNone;
  end;

var
  Form1: TForm1;
  drukarka: textfile;

implementation

uses Unit2, Unit3, Unit6, Unit8;

{$R *.DFM}

procedure TForm1.kWx;
begin
 pisz (ENTER+weather+ENTER+'GA+?'+ENTER);
end;

procedure TForm1.kTgm;
begin
 pisz(ENTER+'ENTER MSG AND END WITH KKKK+'+ENTER+'MSG+?'+ENTER);
 stan:=15; //wysylanie obs
end;

procedure TForm1.kOpr;
begin
 pisz(ENTER+'OPERATOR'S ASSISTANCE IS NOT AVAILABLE IN SIMULATOR'
      +ENTER+'GA+?'+ENTER);
end;

procedure TForm1.kObs;
begin
```

```
 pisz(ENTER+'ENTER MSG AND END WITH KKKK+'+ENTER+'MSG+?'+ENTER);
 stan:=14; //wysylanie obs
end;

procedure TForm1.kTlx(komenda: string);
var
 i: integer;
begin
 i:=pos('TLX',komenda);
 if ((length(komenda)-i)<3) then begin
  kNone;
  abort;
 end;
 pisz(ENTER+'ENTER MSG AND END WITH KKKK+'+ENTER+'MSG+?'+ENTER);
 stan:=14; //wysylanie obs
end;

procedure TForm1.kNav;
begin
 delay(random(4));
 pisz (ENTER);
 pisz (OstrzezeniaNawigacyjne);
 delay (2);
 pisz (ENTER);
 delay (1);
 pisz ('GA+?'+ENTER);
 JestMessage:=false;
end;

procedure TForm1.kMsg;
begin
 if JestMessage then begin
  delay(random(4));
  pisz (ENTER+'MOM PSE'+ENTER);
  delay (4);
  pisz (PrzykladowyMessage);
  delay (2);
  pisz (ENTER);
  delay (1);
  pisz ('GA+?'+ENTER);
  JestMessage:=false;
```

```
 end;
end;

procedure TForm1.kMed;
begin
 pisz ('MEDICAL ASSISTANCE IS NOT AVAILABLE IN THIS SIMULATOR');
 pisz ('GA+?'+ENTER);
end;

procedure TForm1.kFreq;
begin
 delay(2);
 pisz(ENTER+'ENTER MSG AND END WITH KKKK+'+ENTER);
 stan:=13;//czyli po prostu freq
end;

procedure TForm1.kDirtlx(komenda: string);
var
 i: integer;
 pasuje: boolean;
begin
 pasuje:=false;
 for i:=0 to (length(NumeryAbonentow)-1) do begin
  if (pos(IntToStr(NumeryAbonentow[i]),komenda)<>0) then begin
   pasuje:=true;
   ZnamiennikRozmowcy:=ZnamiennikiAbonentow[i];
  end;
 end;
 if (pasuje) then begin
  delay(3);
  pisz(ENTER+'MOM'+ENTER);
  delay (random(6));
  pisz_czas;
  delay (random(3));
  timer1.tag:=0;
  timer1.enabled:=true;
  pisz(ZnamiennikRozmowcy+ENTER);
  delay(3);
  pisz('MSG+?'+ENTER);
  stan:=12; //polaczenie z abonentem z ladu
 end
```

```
 else begin
  delay (3);
  pisz(ENTER+'ABS'+ENTER+'GA+?'+ENTER);
 end;
end;

procedure TForm1.kBrk;
begin
 pisz(ENTER+'-------DISCONNECTED-------'+ENTER);
 stan:=0;
 form3.StatusBar1.Panels[0].text:='Disconnected';
 SpeedButton1.Enabled:=true;
 SpeedButton2.Enabled:=true;
 SpeedButton5.Enabled:=false;
 SpeedButton6.Enabled:=false;
 SpeedButton7.Enabled:=true;
end;

procedure TForm1.kAmv;
begin
 pisz(ENTER+'ENTER MSG AND END WITH KKKK+'+ENTER+'MSG+?'+ENTER);
 stan:=11; //wysylanie amveru
end;


procedure TForm1.kHelp;
var
 i: integer;
begin
 pisz(ENTER+'LIST OF COMMANDS AVAILABLE'+ENTER);
 for i:=0 to 19 do begin
  if DostepneKomendy[PolaczonyZeStacjaNr-1][i+1]='t' then
     pisz (OpisPolecen[i]+ENTER);
 end;
 pisz ('GA+?'+ENTER);
end;

procedure TForm1.kNone;
begin
 pisz(ENTER+'THIS FACILITY IS NOT AVAILABLE, TYPE HELP+'+ENTER+'GA+?'+ENTER);
end;
```

```
procedure TForm1.over_1;
var
 komenda: string;
 i, pol: integer;
begin;
 pol:=255;
 if speedbutton8.down then write(drukarka, bufor+'+'+ENTER);
 komenda:=bufor;
 for i:=(length(polecenia)-1) downto 0 do begin
  if pos(polecenia[i], komenda)<>0 then pol:=i;
 end;
 case pol of
  0: kAmv;
  1: kDirtlx(komenda);
  2: kFreq;
  3: kHelp;
  6: kMed;
  7: kMsg;
  8: kNav;
  9: kObs;
  10: kOpr;
  11: kObs;//tak naprawde to jest POS, tylko ze to i tak jak OBS
  12: kObs;//a to jest RTL
  16: kTgm;
  17: kTlx(komenda);//to jest TLX
  18: kOpr;//to jest URG
  19: kWx;
  20: kBrk;
 else
  kNone;
 end;
 bufor:='';
end;

procedure TForm1.over_7;
var
 czestotliwosc, selcall_, wiersz,a: string;
begin
 if speedbutton8.down then write(drukarka, bufor+'+'+ENTER);
 if (stan=7) then begin
```

```
// ja mowie - rozmawiając z drugim statkiem
//zmien z mowiacego na sluchajacy
stan:=6;
form3.StatusBar1.Panels[4].text:='Listening';
wiersz:=bufor;
pisz(ENTER);
// hmmm, a moze to bylo brk?
if (pos('BRK+', bufor)<>0) then begin
 // czestotliwosc
 czestotliwosc:=FloatToStr(Tx);
 if length(czestotliwosc)>7 then czestotliwosc:='1000   ' else begin
  while (length(czestotliwosc)<7) do begin
   czestotliwosc:=czestotliwosc+' ';
  end;
 end;
 // numer selcall wywolywanej stacji
 selcall_:=IntToStr(selcall);
 if length(selcall_)>7 then selcall_:='10000  ' else begin
  while (length(selcall_)<7) do begin
   selcall_:=selcall_+' ';
  end;
 end;
 a:=czestotliwosc+selcall_+'@BRK!';
 clientsocket1.Socket.SendText(czestotliwosc+selcall_+'@BRK!');
 kBrk;
 {form3.StatusBar1.Panels[1].text:='Disconnected';
 form3.StatusBar1.Panels[4].text:='';
 SpeedButton1.Enabled:=true;
 SpeedButton2.Enabled:=true;
 SpeedButton5.Enabled:=false;
 SpeedButton6.Enabled:=false;}
end else begin
 // czestotliwosc
 czestotliwosc:=FloatToStr(Tx);
 if length(czestotliwosc)>7 then czestotliwosc:='1000   ' else begin
  while (length(czestotliwosc)<7) do begin
   czestotliwosc:=czestotliwosc+' ';
  end;
 end;
 // numer selcall wywolywanej stacji
 selcall_:=IntToStr(selcall);
```

```
   if length(selcall_)>7 then selcall_:='10000  ' else begin
    while (length(selcall_)<7) do begin
     selcall_:=selcall_+' ';
    end;
   end;
   clientsocket1.Socket.SendText(czestotliwosc+selcall_+wiersz);
 end;
 end;
 bufor:='';
end;

procedure TForm1.over_11;
begin
 if speedbutton8.down then write(drukarka, bufor+'+'+ENTER);
 if (pos('KKKK+', bufor)<>0) then begin
  pisz(ENTER+ZnamiennikRozmowcy+ENTER+MojZnamiennik+ENTER+ENTER+
       'AMVER MSG ACCEPTED'+ENTER+'MSG REF : 016547'+ENTER+'GA+? ');
  stan:=1;
 end;
 bufor:='';
end;

procedure TForm1.over_12;
begin
 if speedbutton8.down then write(drukarka, bufor+'+'+ENTER);
 if (pos('KKKK+', bufor)<>0) then begin
  pisz_czas;
  pisz (ZnamiennikRozmowcy+ENTER);
  pisz ('DURATION OF CALL: ');
  timer1.enabled:=false;
  pisz (IntToStr(trunc(timer1.tag/60))+':'+IntToStr
       (timer1.tag-(60*(trunc(timer1.tag/60))))+ENTER);
  timer1.tag:=0;
  stan:=1;
 end;
 bufor:='';
end;

procedure TForm1.over_13;
begin
 if speedbutton8.down then write(drukarka, bufor+'+'+ENTER);
```

```
 delay(random(4));
 pisz (ENTER+ZnamiennikRozmowcy+ENTER);
 pisz (MojZnamiennik+ENTER);
 delay (3);
 pisz ('MSG ACCEPTED'+ENTER+'MSG REF : 016547'+ENTER+'GA+?'+ENTER);
 stan:=1;
end;


procedure TForm1.over_14;
begin
 if speedbutton8.down then write(drukarka, bufor+'+'+ENTER);
 delay(random(4));
 pisz (ENTER+ZnamiennikRozmowcy+ENTER);
 pisz (MojZnamiennik+ENTER);
 delay (3);
 pisz ('MSG ACCEPTED'+ENTER+'MSG REF : 016547'+ENTER+'GA+?'+ENTER);
 stan:=1;
end;

procedure TForm1.over_15;
begin
 if speedbutton8.down then write(drukarka, bufor+'+'+ENTER);
 delay(random(4));
 pisz (ENTER+ZnamiennikRozmowcy+ENTER);
 pisz (MojZnamiennik+ENTER);
 delay (3);
 pisz ('2000 05 11 23:17    1 TGM REF:A02GQ'+ENTER+'GA+?'+ENTER);
 stan:=1;
end;

procedure TForm1.pisz(co: string);
var i:cardinal;
begin
 for i:=1 to length(co) do begin
  if (co[i]<>chr(10)) then begin
   Form3.Memo1.seltext:=co[i];
  end else begin
   Form3.Memo1.seltext:=chr(13)+chr(10);
  end;
  if speedbutton8.down then write(drukarka, co[i]);
```

```
  Form3.Memo1.refresh;
  Form3.Memo1.text;
  delay(1);
  application.ProcessMessages;
  if (application.terminated) then abort;
 end;
end;

procedure TForm1.pisz_czas;
 var
  SystemTime:TSystemTime;
begin
 GetLocalTime(SystemTime);
 pisz(chr(10));
 pisz(DateTimeToStr(SystemTimeToDateTime(SystemTime)));
 pisz(chr(10));
end;

procedure TForm1.delay(czas: integer);
begin
 sleep(czas*Zwolnienie);
end;

procedure TForm1.Editor1Click(Sender: TObject);
begin
 form2.windowstate:=wsmaximized;
end;

procedure TForm1.Radiotelex1Click(Sender: TObject);
begin
 form3.windowstate:=wsmaximized;
end;

procedure TForm1.newtelex1Click(Sender: TObject);
begin
 form2.Memo1.text:='';
 form2.filename:='?';
 form2.caption:='editor - new file';
end;

procedure TForm1.savetelex1Click(Sender: TObject);
```

```
begin
 if (form2.filename='?') then begin
  if (savedialog1.execute) then begin
   form2.filename:=savedialog1.filename;
   form2.memo1.Lines.savetofile(savedialog1.filename);
   form2.caption:='editor - "'+form2.filename+'"';
  end;
 end
 else begin
  form2.memo1.lines.savetofile(form2.filename);
 end;
end;

procedure TForm1.savetelexas1Click(Sender: TObject);
begin
 if (savedialog1.execute) then begin
  form2.filename:=savedialog1.filename;
  form2.memo1.Lines.savetofile(savedialog1.filename);
  form2.caption:='editor - "'+form2.filename+'"';
 end;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
 assignfile(drukarka,'lpt1');
 rewrite (drukarka);
 DecimalSeparator:='.';
 TEST:=false;
 JestMessage:=true;
 sciezka:='C:\telex\';
 stan:=0;
 rx:=2815.5;
 tx:=1971.5;
 selcall:=0480;
 connected:=false;
 randomize;
 enter:=chr(10);
 PolaczonyZCentrala:=false;
 SetLength(polecenia,21);
 polecenia[0]:='AMV';
 polecenia[1]:='DIRTLX';
```

```
 polecenia[2]:='FREQ';
 polecenia[3]:='HELP';
 polecenia[4]:='INF';
 polecenia[5]:='MAN';
 polecenia[6]:='MED';
 polecenia[7]:='MSG';
 polecenia[8]:='NAV';
 polecenia[9]:='OBS';
 polecenia[10]:='OPR';
 polecenia[11]:='POS';
 polecenia[12]:='RTL';
 polecenia[13]:='STA';
 polecenia[14]:='STS';
 polecenia[15]:='SVC';
 polecenia[16]:='TGM';
 polecenia[17]:='TLX';
 polecenia[18]:='URG';
 polecenia[19]:='WX';
 polecenia[20]:='BRK';


 SetLength(OpisPolecen,20);
 OpisPolecen[0]:='AMV+      SENDING AMVER MESSAGES';
 OpisPolecen[1]:='DIRTLX+      DIRECT CONNECTION TO TELEX SUBSCRIBER';
 OpisPolecen[2]:='FREQ+      INDICATING SHIPS WATCH FREQUENCY,
SCHEDULES, TIME INTERVALS AND CHANNELS
ON WHICH SHIPS ARE STANDING BY FOR ARQ,
FEC OR SELFEC';
 OpisPolecen[3]:='HELP+      A LIST OF COMMANDS AVAILABLE
(LIKE ONE YOU READ NOW';
 OpisPolecen[4]:='INF+      REQUESTING RADIOTELEX INFORMATION SERVICES';
 OpisPolecen[5]:='MAN+      MANUAL CONNECTION';
 OpisPolecen[6]:='MED+      RADIOMEDICAL CONNECTION';
 OpisPolecen[7]:='MSG+      REQUESTING TRAFFIC';
 OpisPolecen[8]:='NAV+      REQUESTING NAVIGATIONAL WARNINGS';
 OpisPolecen[9]:='OBS+      SENDING SHIPS WEATHER REPORT';
 OpisPolecen[10]:='OPR+      OPERATORS ASSISTANCE';
 OpisPolecen[11]:='POS+      SENDING SHIPS POSITION REPORT';
 OpisPolecen[12]:='RTL+      SENDING A RADIOTELEX LETTER';
 OpisPolecen[13]:='STA+      REQUESTING STATUS OF STORE AND FORWARD MESSAGES';
 OpisPolecen[14]:='STS+      SENDING A SHIP TO SHIP MESSAGE';
```

```
 OpisPolecen[15]:='SVC+      SENDING A SERVICE MESSAGE';
 OpisPolecen[16]:='TGM+      SENDING A RADIOTELEGRAM';
 OpisPolecen[17]:='TLX+      SENDING A STORE AND FORWARD MESSAGE';
 OpisPolecen[18]:='URG+      REQUESTING OPERATORS ASSISTANCE';
 OpisPolecen[19]:='WX+      REQUESTING WEATHER WARNINGS';
end;

procedure TForm1.Frequency1Click(Sender: TObject);
begin
 if (stan=0) then Form6.visible:=true;
end;

procedure TForm1.ARQ1Click(Sender: TObject);
 var
  PasujeStacjaBrzegowa: boolean;
  i,j,k,l: integer;
  znam_, czestotliwosc, selcall_, a: string;
begin
 PasujeStacjaBrzegowa:=false;
 for k:=1 to ((length(Czestotliwosci)+1) div 4) do begin
  i:=(k-1)*4;
  if
    ((StrToFloat(czestotliwosci[i])=(Rx)) AND
     (StrToFloat(czestotliwosci[i+1])  =(Tx)) AND
     (StrToInt  (czestotliwosci[i+2])=(Selcall)) AND
     (stan=0))
  then begin
   PasujeStacjaBrzegowa:=true;
   l:=i;
  end;
 end;
 if PasujeStacjaBrzegowa then begin
  connected:=true;
  j:=StrToInt(czestotliwosci[l+3]);
  ZnamiennikStacjiBrzegowej:=ZnamiennikiStacjiBrzegowych[j-1];
  DostepneAktualnieKomendy:=DostepneKomendy[j-1];
  ZnamiennikRozmowcy:=ZnamiennikStacjiBrzegowej;
  PolaczonyZeStacjaNr:=j;
  //zaczynamy wypisywanie na ekran
  delay(random(30));
  // napisz ze jestesmy polaczeni
```

```
 form3.StatusBar1.Panels[0].text:='Connected';
 SpeedButton1.Enabled:=false;
 SpeedButton2.Enabled:=false;
 SpeedButton5.Enabled:=true;
 SpeedButton6.Enabled:=true;
 SpeedButton7.Enabled:=false;
 form3.StatusBar1.refresh;
 {wypisuje answerback stacji}
 pisz(ZnamiennikStacjiBrzegowej+enter);
 form1.refresh;
 delay(4);
 {wypisuje nasz answerback}
 pisz(MojZnamiennik+enter);
 form1.refresh;
 {znowu czeka}
 delay(4);
 {no i na koniec pisze "GA+"}
 pisz('GA+? '+enter);
 // napisz ze gadamy
 form3.StatusBar1.Panels[4].text:='Speaking';
 form3.StatusBar1.refresh;
 // stan = 1 czyli "polaczony ze stacja brzegowa, normalny tryb"
 stan:=1;
end
else begin
 // jesli to nie jest zadna stacja brzegowa, to ...
 if (PolaczonyZCentrala) then begin
  // czestotliwosc
  czestotliwosc:=FloatToStr(Tx);
  if length(czestotliwosc)>7 then czestotliwosc:='1000   ' else begin
   while (length(czestotliwosc)<7) do begin
    czestotliwosc:=czestotliwosc+' ';
   end;
  end;
  // numer selcall wywolywanej stacji
  selcall_:=IntToStr(selcall);
  if length(selcall_)>7 then selcall_:='10000  ' else begin
   while (length(selcall_)<7) do begin
    selcall_:=selcall_+' ';
   end;
  end;
```

```
   znam_:=(MojZnamiennik);
   // moj znamiennik
   if length(znam_)>20 then znam_:='10000  ' else begin
    while (length(znam_)<20) do begin
     znam_:=znam_+' ';
    end;
   end;
   a:=czestotliwosc+selcall_+'@hop!'+znam_;
   clientsocket1.Socket.SendText(a);
   // wypisz moj znamiennik
   {pisz(MojZnamiennik+enter);
   form1.refresh;    }
   // zmien stan
   stan:=5; // 5 oznacza - czeka na odpowiedz drugiego statku,
            // zeby byc polaczonym
  end;
 end;
end;

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
 ARQ1Click(Self);
end;

procedure TForm1.ClientSocket1Read(Sender: TObject;
  Socket: TCustomWinSocket);
var
 i:integer;
 znam_, a, jego_znamiennik, uslyszalem,
 czestotliwosc, selcall_, komenda: string;
begin
form3.StatusBar1.Panels[6].text:='on read';////////////////
 uslyszalem:=socket.ReceiveText;
 if TEST then begin
  uslyszalem:=Test_ReceiveText;
  TEST:=false;
 end;
 // jesli stan "0"
 if (stan=0) then begin
  czestotliwosc:=copy(uslyszalem,1,7);
  selcall_:=copy(uslyszalem,8,7);
```

```
komenda:=copy(uslyszalem,15,5);
// usun spacje z czestotliwosci
for i:=(length(czestotliwosc)-1) downto 1 do begin
 if (czestotliwosc[i+1]=' ') then begin
  czestotliwosc:=copy(czestotliwosc,1,i);
 end
 else break;
end;
// usun spacje z selkola
for i:=(length(selcall_)-1) downto 1 do begin
 if (selcall_[i+1]=' ') then begin
  selcall_:=copy(selcall_,1,i);
 end
 else break;
end;

if
((komenda='@hop!')                    AND
 (StrToFloat(czestotliwosc)=Rx)       AND
 (StrToInt(selcall_)=MojNumerSelcall))
                                           then begin
 jego_znamiennik:=copy(uslyszalem, 20, 20);
 pisz (jego_znamiennik+ENTER);
 ZnamiennikRozmowcy:=jego_znamiennik;
 // aktualny selcall zmien na selcall tego, kto cie wywolal
 selcall:=StrToInt(copy(jego_znamiennik,1,5));
 form3.StatusBar1.Panels[3].text:='Selcall number: '+IntToStr(selcall);
 // odpowiedz mu. polaczenie bedzie nawiazane
 // a ja bede w stanie "6" - "polaczony ze statkiem i slucha"

 // czestotliwosc
 czestotliwosc:=FloatToStr(Tx);
 if length(czestotliwosc)>7 then czestotliwosc:='1000   ' else begin
  while (length(czestotliwosc)<7) do begin
    czestotliwosc:=czestotliwosc+' ';
  end;
 end;
 // numer selcall wywolywanej stacji
 selcall_:=IntToStr(selcall);
 if length(selcall_)>7 then selcall_:='10000  ' else begin
  while (length(selcall_)<7) do begin
```

```
    selcall_:=selcall_+' ';
   end;
  end;
  znam_:=(MojZnamiennik);
  // moj znamiennik
  if length(znam_)>20 then znam_:='10000  ' else begin
   while (length(znam_)<20) do begin
     znam_:=znam_+' ';
   end;
  end;
 // a teraz chwila przerwy
 delay(30);
 // teraz wyslij do niego odpowiedz
 a:=czestotliwosc+selcall_+'@odp!'+znam_;
 clientsocket1.Socket.SendText(a);
 stan:=6;
 form3.StatusBar1.Panels[4].text:='Listening';
 pisz(MojZnamiennik+ENTER);
 pisz ('GA+?');
 form3.StatusBar1.Panels[0].text:='Connected';
 SpeedButton2.Enabled:=false;
 SpeedButton1.Enabled:=false;
 SpeedButton5.Enabled:=false;
 SpeedButton6.Enabled:=false;
 end;
 //a moze to FEC?
 if
 ((komenda='@fec!')                    AND
  (StrToFloat(czestotliwosc)=Rx))
 then begin
  pisz (copy(uslyszalem, 20, length(uslyszalem)-17));
 end;
end;
// jesli stan "5" - czyli jesli czekam na polaczenie ze statkiem
if (stan=5) then begin
 czestotliwosc:=copy(uslyszalem,1,7);
 selcall_:=copy(uslyszalem,8,7);
 komenda:=copy(uslyszalem,15,5);
 // usun spacje z czestotliwosci
 for i:=(length(czestotliwosc)-1) downto 1 do begin
  if (czestotliwosc[i+1]=' ') then begin
```

```
   czestotliwosc:=copy(czestotliwosc,1,i);
  end
  else break;
 end;
 // usun spacje z selkola
 for i:=(length(selcall_)-1) downto 1 do begin
  if (selcall_[i+1]=' ') then begin
   selcall_:=copy(selcall_,1,i);
  end
  else break;
 end;

 if
 ((komenda='@odp!')                    AND
  (StrToFloat(czestotliwosc)=Rx)       AND
  (StrToInt(selcall_)=MojNumerSelcall))
                                              then begin
  jego_znamiennik:=copy(uslyszalem, 20, 20);
  {wypisuje answerbacki stacji}
  pisz(MojZnamiennik+ENTER);
  form1.refresh;
  delay(4);
  pisz (jego_znamiennik+ENTER);
  pisz ('GA+?'+ENTER);
  ZnamiennikRozmowcy:=jego_znamiennik;
 // zmien stan na "7" - czyli "polaczony ze statkiem - ja mowie"
 stan:=7;
 form3.StatusBar1.Panels[4].text:='Speaking';
 end;
end;
// jesli stan "6" - ja slucham od drugiego statku
if (stan=6) then begin
 czestotliwosc:=copy(uslyszalem,1,7);
 selcall_:=copy(uslyszalem,8,7);
 komenda:=copy(uslyszalem,15,1);
 // usun spacje z czestotliwosci
 for i:=(length(czestotliwosc)-1) downto 1 do begin
  if (czestotliwosc[i+1]=' ') then begin
   czestotliwosc:=copy(czestotliwosc,1,i);
  end
  else break;
```

```
  end;
  // usun spacje z selkola
  for i:=(length(selcall_)-1) downto 1 do begin
   if (selcall_[i+1]=' ') then begin
    selcall_:=copy(selcall_,1,i);
   end
   else break;
  end;
  // a moze to BRK - rozlaczenie?
  if
  ((komenda='@')                    AND
   (copy(uslyszalem,15,5)='@BRK!')        AND
   (StrToFloat(czestotliwosc)=Rx)       AND
   (StrToInt(selcall_)=MojNumerSelcall))
                                              then begin
   kBrk;
  end;
  if
   (
   (komenda<>'@') AND
   (StrToFloat(czestotliwosc)=Rx) AND
   (StrToInt(selcall_)=MojNumerSelcall)
   )
  then begin
   pisz(copy(uslyszalem,15,length(uslyszalem)-14));
   pisz(ENTER);
   stan:=7;
   form3.StatusBar1.Panels[4].text:='Speaking';
  end;
 end;
end;

procedure TForm1.ClientSocket1Connect(Sender: TObject;
  Socket: TCustomWinSocket);
begin
 form3.StatusBar1.Panels[6].text:='on connect';
end;

procedure TForm1.ClientSocket1Connecting(Sender: TObject;
  Socket: TCustomWinSocket);
begin
```

```
form3.StatusBar1.Panels[6].text:='on connecting';//////////
end;

procedure TForm1.ClientSocket1Lookup(Sender: TObject;
  Socket: TCustomWinSocket);
begin
form3.StatusBar1.Panels[6].text:='on lookup';/////////////
end;

procedure TForm1.laczzcentrala1Click(Sender: TObject);
begin
  form1.clientsocket1.Address :=form1.IPCentrali;
  form1.clientsocket1.active:=true;
  form1.PolaczonyZCentrala:=true;
end;

procedure TForm1.socketonread1Click(Sender: TObject);
begin
 Test_ReceiveText:=(InputBox('-----TEST-------',
 'tekst otrzymany rzekomo przez soket: ','2815.5 0000000@fec!TEST'));
 TEST:=true;
 ClientSocket1Read(self,ClientSocket1.Socket);
end;

procedure TForm1.stan1Click(Sender: TObject);
begin
 ShowMessage(IntToStr(stan));
end;

procedure TForm1.FEC1Click(Sender: TObject);
var
 a, czestotliwosc: string;
begin
 // czestotliwosc
 czestotliwosc:=FloatToStr(Tx);
 if length(czestotliwosc)>7 then czestotliwosc:='1000   ' else begin
  while (length(czestotliwosc)<7) do begin
   czestotliwosc:=czestotliwosc+' ';
  end;
 end;
 a:=czestotliwosc+'0000000'+'@fec!'+form2.memo1.text;
```

```
 clientsocket1.Socket.SendText(a);
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
 timer1.tag:=timer1.tag+1;
end;

procedure TForm1.Whoareyou1Click(Sender: TObject);
begin
 delay(15);
 pisz (ZnamiennikRozmowcy);
 pisz(ENTER+'GA+?'+ENTER);
end;

procedure TForm1.Thisis1Click(Sender: TObject);
begin
 delay(2);
 pisz (MojZnamiennik);
 pisz(ENTER+'GA+?'+ENTER);
end;

procedure TForm1.Sendfile1Click(Sender: TObject);
var
 i, pos1, pos2:integer;
 klawisz: char;
 czestotliwosc, selcall_: string;
begin
 if (OpenDialog1.Execute) then begin
  Memo1.Lines.LoadFromFile(OpenDialog1.FileName);
  //W trybie 1 - rozmowa ze stacja brzegowa
  if (
      (stan=1) OR
      (stan=11) OR
      (stan=12) OR
      (stan=13) OR
      (stan=14) OR
      (stan=15)
      ) then begin
   pisz(Memo1.Text);
   if (pos('+',Memo1.Text)<>0) then begin
```

```
    for i:=1 to length(Memo1.Text) do begin
     if (Memo1.Text[length(Memo1.Text)-i]='+') then begin
      pos1:=i;
      break;
     end;
    end;
    for i:=pos1 to length(Memo1.Text) do begin
     if (Memo1.Text[length(Memo1.Text)-i]=' ') then begin
      pos2:=i;
      break;
     end;
    end;
    bufor:=copy(Memo1.Text,pos1,pos2-pos1);
    klawisz:='+';
    Form3.Memo1KeyPress(self,klawisz);
   end;
  end;
  //A tu bedzie tryb 7
  // czestotliwosc
  czestotliwosc:=FloatToStr(Tx);
  if length(czestotliwosc)>7 then czestotliwosc:='1000   ' else begin
   while (length(czestotliwosc)<7) do begin
    czestotliwosc:=czestotliwosc+' ';
   end;
  end;
  // numer selcall wywolywanej stacji
  selcall_:=IntToStr(selcall);
  if length(selcall_)>7 then selcall_:='10000  ' else begin
   while (length(selcall_)<7) do begin
    selcall_:=selcall_+' ';
   end;
  end;
  clientsocket1.Socket.SendText(czestotliwosc+selcall_+Memo1.Text);
 end;
end;

procedure TForm1.Sendtext1Click(Sender: TObject);
var
 i, pos1, pos2:integer;
 klawisz: char;
 czestotliwosc, selcall_: string;
```

```
begin
 if (Form2.Memo1.Text<>'') then begin
  Memo1.Text:=Form2.Memo1.Text;
  //W trybie 1 - rozmowa ze stacja brzegowa
  if (
      (stan=1) OR
      (stan=11) OR
      (stan=12) OR
      (stan=13) OR
      (stan=14) OR
      (stan=15)
      ) then begin
   pisz(Memo1.Text);
   if (pos('+',Memo1.Text)<>0) then begin
    for i:=1 to length(Memo1.Text) do begin
     if (Memo1.Text[length(Memo1.Text)-i]='+') then begin
      pos1:=i;
      break;
     end;
    end;
    for i:=pos1 to length(Memo1.Text) do begin
     if (Memo1.Text[length(Memo1.Text)-i]=' ') then begin
      pos2:=i;
      break;
     end;
    end;
    bufor:=copy(Memo1.Text,pos1,pos2-pos1);
    klawisz:='+';
    Form3.Memo1KeyPress(self,klawisz);
   end;
  end;
  //A tu bedzie tryb 7
  // czestotliwosc
  czestotliwosc:=FloatToStr(Tx);
  if length(czestotliwosc)>7 then czestotliwosc:='1000   ' else begin
   while (length(czestotliwosc)<7) do begin
    czestotliwosc:=czestotliwosc+' ';
   end;
  end;
  // numer selcall wywolywanej stacji
  selcall_:=IntToStr(selcall);
```

```
  if length(selcall_)>7 then selcall_:='10000  ' else begin
   while (length(selcall_)<7) do begin
    selcall_:=selcall_+' ';
   end;
  end;
  clientsocket1.Socket.SendText(czestotliwosc+selcall_+Memo1.Text);
 end;
end;
procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
 Radiotelex1Click(self);
end;

procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
 Editor1Click(self);
end;

procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
 Sendfile1Click(self);
end;

procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
 Sendtext1Click(self);
end;

procedure TForm1.selcall1Click(Sender: TObject);
begin
 MojNumerSelcall:=StrToInt(InputBox('-----TEST-------',
  'nowy numer selcall: ','1'));
 MojZnamiennik:=(InputBox('-----TEST-------',
  'nowy znamiennik: ','00001 PROGRAM PC'));
 form3.caption:='radiotelex   '+IntToStr(form1.MojNumerSelcall);
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
 FEC1Click(self);
end;
```

```
procedure TForm1.AddressBook1Click(Sender: TObject);
begin
 Form8.Show;
end;

procedure TForm1.SpeedButton7Click(Sender: TObject);
begin
 AddressBook1Click(self);
end;

procedure TForm1.stan2Click(Sender: TObject);
begin
 stan:=StrToInt(InputBox('------TEST--------','stan =?','0'));

end;

procedure TForm1.Over1Click(Sender: TObject);
begin
 pisz ('+');
 case stan of
  1: over_1;
  7: over_7;
  11: over_11;
  12: over_12;
  13: over_13;
  14: over_14;
  15: over_15;
 end;
end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
 try
  closefile(drukarka);
 except
 end;
end;

procedure TForm1.SpeedButton9Click(Sender: TObject);
begin
```

```
 form3.memo1.text:='';
end;

procedure TForm1.SpeedButton10Click(Sender: TObject);
begin
 if (savedialog1.execute) then begin
  form3.memo1.lines.SaveToFile(savedialog1.filename);
 end;
end;

end.
```

9.6.2 Unit2.pas

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, Menus, StdCtrls;

type
  TForm2 = class(TForm)
    Memo1: TMemo;
    procedure FormActivate(Sender: TObject);
    procedure FormDeactivate(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Memo1KeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
    filename: string;
  end;

const
 dostepne_znaki=['A','B','C','D','E','F','G',
                 'H','I','J','K','L','M','N',
                 'O','P','Q','R','S','T','U',
                 'V','W','X','Y','Z','0','1',
                 '2','3','4','5','6','7','8',
                 '9','@','#','%','(',')','/',
                 '?','"','+',#10,#13];

var
  Form2: TForm2;

implementation

uses Unit1, Unit3;

{$R *.DFM}
```

```
procedure TForm2.FormActivate(Sender: TObject);
begin
 form1.mainmenu1.Items.Items[0].enabled:=true;
 form1.SpeedButton1.enabled:=false;
 form1.SpeedButton2.enabled:=false;
 form1.SpeedButton3.enabled:=false;
 form1.SpeedButton4.enabled:=true;
end;

procedure TForm2.FormDeactivate(Sender: TObject);
begin
 form1.mainmenu1.Items.Items[0].enabled:=false;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin
 filename:='?';
end;

procedure TForm2.Memo1KeyPress(Sender: TObject; var Key: Char);
begin
 key:=uppercase(key)[1];
 if (not ((key in dostepne_znaki) or (key=chr(8)))) then key:=' ';
end;

end.
```

9.6.3 Unit3.pas

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, ComCtrls, StdCtrls;

type
  TForm3 = class(TForm)
    StatusBar1: TStatusBar;
    Memo1: TRichEdit;
    procedure FormCreate(Sender: TObject);
    procedure Memo1KeyPress(Sender: TObject; var Key: Char);
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    poprzedniKlawisz: char;
    status: byte;
  end;
const
 dostepne_znaki=['A','B','C','D','E','F','G','H',
                 'I','J','K','L','M','N','O','P',
                 'Q','R','S','T','U','V','W','X',
                 'Y','Z','0','1','2','3','4','5',
                 '6','7','8','9','@','#','%','(',
                 ')','/','?','"','+',#10,#13];

var
  Form3: TForm3;

implementation

uses Unit1;

{$R *.DFM}
```

```
procedure TForm3.FormCreate(Sender: TObject);
begin
 status:=0;
end;

procedure TForm3.Memo1KeyPress(Sender: TObject; var Key: Char);
begin

 if (
     (key='+') AND
     (form1.stan = 11)
    )
     then begin
      Memo1.Seltext:='+';
      form1.bufor:=form1.bufor+'+';
      key:=' ';
      form1.over_11;
     end;

 if (
     (key='+') AND
     (form1.stan = 12)
    )
     then begin
      Memo1.Seltext:='+';
      form1.bufor:=form1.bufor+'+';
      key:=' ';
      form1.over_12;
     end;

 if (
     (key='+') AND
     (form1.stan = 13)
    )
     then begin
      Memo1.Seltext:='+';
      form1.bufor:=form1.bufor+'+';
      key:=' ';
      form1.over_13;
     end;
```

```
if (
    (key='+') AND
    (form1.stan = 14)
   )
    then begin
     Memo1.Seltext:='+';
     form1.bufor:=form1.bufor+'+';
     key:=' ';
     form1.over_14;
    end;

if (
    (key='+') AND
    (form1.stan = 15)
   )
    then begin
     Memo1.Seltext:='+';
     form1.bufor:=form1.bufor+'+';
     key:=' ';
     form1.over_15;
    end;

if (
    (key='?') AND
    (PoprzedniKlawisz='+') AND
    (form1.stan = 7)
   )
    then begin
     Memo1.Seltext:='?';
     form1.bufor:=form1.bufor+'?';
     key:=' ';
     form1.over_7;
    end;

if (
    (key='+') AND
    (form1.stan=1)
    )
    then begin
     Memo1.Seltext:='+';
     key:=' ';
```

```
      form1.over_1;
     end;
 //showmessage(inttostr(ord(key)));   //debug
 key:=uppercase(key)[1];
 if (not ((key in dostepne_znaki) or (key=chr(8)))) then key:=' ';
 poprzedniKlawisz:=key;
 form1.bufor:=form1.bufor+key;
 if(ord(key)=13) then form1.bufor:=form1.bufor+chr(10);
end;

procedure TForm3.FormActivate(Sender: TObject);
begin
 form1.mainmenu1.Items.Items[0].enabled:=false;
 if (form1.stan=0) then begin
  form1.SpeedButton1.enabled:=true;
  form1.SpeedButton2.enabled:=true;
 end;
 form1.SpeedButton3.enabled:=true;
 form1.SpeedButton4.enabled:=false;
 form1.SpeedButton5.enabled:=(form1.stan<>0);
 form1.SpeedButton6.enabled:=(form1.stan<>0);

end;

end.
```

9.6.4 Unit4.pas

```
unit Unit4;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, FileCtrl;

type
  TForm4 = class(TForm)
    config: TMemo;
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form4: TForm4;

implementation

uses Unit5, Unit1, Unit3;

{$R *.DFM}

procedure TForm4.FormCreate(Sender: TObject);
 var
  i: integer;
  pos, koniec: integer;
begin
 //sprobuj wczytac ksiazke adresowa
 try
  Form1.KsiazkaAdresowa.lines.loadfromfile(form1.sciezka+'adresy.txt');
 except
  ForceDirectories(form1.sciezka);
  Form1.KsiazkaAdresowa.Lines.SaveToFile(form1.sciezka+'adresy.txt');
 end;
```

```
//sprawdzamy, czy program nie jest uruchamiany pierwszy raz
try
 config.lines.loadfromfile(form1.sciezka+'config.txt');
except
 form5.visible:=true;
end;
//teraz z pliku config.txt spisujemy do tablic ustawienia
//najpierw znajdzmy moj numer selcall
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# moj numer selcall') then begin
  pos:=i;
  break;
 end;
end;
form1.MojNumerSelcall:=StrToInt(config.Lines[pos+1]);
form3.caption:='radiotelex   '+IntToStr(form1.MojNumerSelcall);
//teraz wyszukujemy moj znamiennik
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# moj znamiennik') then begin
  pos:=i;
  break;
 end;
end;
form1.MojZnamiennik:=config.Lines[pos+1];
// zwolnienie
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# zwolnienie') then begin
  pos:=i;
  break;
 end;
end;
form1.zwolnienie:=StrToInt(config.Lines[pos+1]);
// IP centrali
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# IP centrali') then begin
  pos:=i;
  break;
 end;
end;
form1.IPcentrali:=(config.Lines[pos+1]);
// lacz z centrala
```

```
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# automatycznie lacz z centrala') then begin
  pos:=i;
  break;
 end;
end;
if (config.Lines[pos+1]='tak') then
                  form1.LaczZCentrala:=true
            else form1.LaczZCentrala:=false;
// stacje brzegowe
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# stacje brzegowe') then begin
  pos:=i;
  break;
 end;
end;
for i:=pos+1 to config.Lines.Count do begin;
 if (config.lines[i][1]='#') then begin
  koniec:=i;
  break;
 end;
end;
SetLength(Form1.StacjeBrzegowe,(koniec-pos-1));
for i:=pos+1 to koniec-1 do begin
 form1.StacjeBrzegowe[i-(pos+1)]:=config.lines[i];
end;
//czestotliwosci
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# czestotliwosci') then begin
  pos:=i;
  break;
 end;
end;
for i:=pos+1 to config.Lines.Count do begin;
 if (config.lines[i][1]='#') then begin
  koniec:=i;
  break;
 end;
end;
SetLength(Form1.Czestotliwosci,(koniec-pos-1));
for i:=pos+1 to koniec-1 do begin
```

```
 form1.Czestotliwosci[i-(pos+1)]:=(config.lines[i]);
end;
//znamienniki stacji brzegowych
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# znamienniki stacji brzegowych') then begin
  pos:=i;
  break;
 end;
end;
for i:=pos+1 to config.Lines.Count do begin;
 if (config.lines[i][1]='#') then begin
  koniec:=i;
  break;
 end;
end;
SetLength(Form1.ZnamiennikiStacjiBrzegowych,(koniec-pos-1));
for i:=pos+1 to koniec-1 do begin
 form1.ZnamiennikiStacjiBrzegowych[i-(pos+1)]:=(config.lines[i]);
end;
//dostepne komendy
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# dostepne komendy') then begin
  pos:=i;
  break;
 end;
end;
for i:=pos+1 to config.Lines.Count do begin;
 if (config.lines[i][1]='#') then begin
  koniec:=i;
  break;
 end;
end;
SetLength(Form1.DostepneKomendy,(koniec-pos-1));
for i:=pos+1 to koniec-1 do begin
 form1.DostepneKomendy[i-(pos+1)]:=(config.lines[i]);
end;
//numery abonentów
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# numery abonentow') then begin
  pos:=i;
  break;
```

```
 end;
end;
for i:=pos+1 to config.Lines.Count do begin;
 if (config.lines[i][1]='#') then begin
  koniec:=i;
  break;
 end;
end;
SetLength(Form1.NumeryAbonentow,(koniec-pos-1));
for i:=pos+1 to koniec-1 do begin
 form1.NumeryAbonentow[i-(pos+1)]:=StrToInt(config.lines[i]);
end;
//znamienniki abonentów
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# znamienniki abonentow') then begin
  pos:=i;
  break;
 end;
end;
for i:=pos+1 to config.Lines.Count do begin;
 if (config.lines[i][1]='#') then begin
  koniec:=i;
  break;
 end;
end;
SetLength(Form1.ZnamiennikiAbonentow,(koniec-pos-1));
for i:=pos+1 to koniec-1 do begin
 form1.ZnamiennikiAbonentow[i-(pos+1)]:=(config.lines[i]);
end;
//PrzykladowyMessage
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# przykladowy message') then begin
  pos:=i;
  break;
 end;
end;
for i:=pos+1 to config.Lines.Count do begin;
 if (config.lines[i][1]='#') then begin
  koniec:=i;
  break;
 end;
```

```
end;
for i:=pos+1 to koniec-1 do begin
 form1.PrzykladowyMessage:=form1.PrzykladowyMessage+
                          (config.lines[i])+Form1.ENTER;
end;
//OstrzezeniaNawigacyjne
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# ostrzezenia nawigacyjne') then begin
  pos:=i;
  break;
 end;
end;
for i:=pos+1 to config.Lines.Count do begin;
 if (config.lines[i][1]='#') then begin
  koniec:=i;
  break;
 end;
end;
for i:=pos+1 to koniec-1 do begin
 form1.OstrzezeniaNawigacyjne:=form1.OstrzezeniaNawigacyjne+
                               (config.lines[i])+Form1.ENTER;
end;
//Weather
for i:=0 to config.Lines.Count do begin;
 if (config.lines[i]='# weather') then begin
  pos:=i;
  break;
 end;
end;
for i:=pos+1 to config.Lines.Count do begin;
 if (config.lines[i][1]='#') then begin
  koniec:=i;
  break;
 end;
end;
for i:=pos+1 to koniec-1 do begin
 form1.Weather:=form1.Weather+(config.lines[i])+Form1.ENTER;
end;

// A teraz, jesli trzeba, polacz z centrala
if (form1.LaczZCentrala) then begin
```

```
  form1.clientsocket1.Address :=form1.IPCentrali;
  form1.clientsocket1.active:=true;
  form1.PolaczonyZCentrala:=true;
 end;
end;

end.
```

9.6.5 Unit5.pas

```
unit Unit5;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Mask, FileCtrl;

type
  TForm5 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Memo1: TMemo;
    MaskEdit1: TMaskEdit;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Label3: TLabel;
    Edit2: TEdit;
    CheckBox1: TCheckBox;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure MaskEdit1Change(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation

uses Unit4, Unit1;

{$R *.DFM}

procedure TForm5.Button1Click(Sender: TObject);
```

```
begin
 if (maskedit1.text='00000') then maskedit1.text:='99999';
 form4.config.lines[1]:=maskedit1.text;
 form4.config.Lines[3]:=edit1.text;
 form4.config.Lines[7]:=edit2.text;
 if checkbox1.checked then form4.config.Lines[9]:='tak'
    else form4.config.Lines[9]:='nie';
 form5.visible:=false;
 ForceDirectories(form1.sciezka);
 form4.config.lines.savetofile(form1.sciezka+'config.txt');

end;

procedure TForm5.Button2Click(Sender: TObject);
begin
 form5.visible:=false;
end;

procedure TForm5.MaskEdit1Change(Sender: TObject);
begin
 edit1.text:=maskedit1.text+copy(edit1.text,6,length(edit1.text)-5);
end;

end.
```

9.6.6 Unit6.pas

```
unit Unit6;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, Buttons;

type
  TForm6 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Button1: TButton;
    SpeedButton1: TSpeedButton;
    procedure Button1Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form6: TForm6;

implementation

uses Unit3, Unit1, Unit8;

{$R *.DFM}

procedure TForm6.Button1Click(Sender: TObject);
begin
 Form6.Visible:=false;
 form3.StatusBar1.Panels[1].text:='Rx: '+Form6.Edit1.Text;
```

```
 form3.StatusBar1.Panels[2].text:='Tx: '+Form6.Edit2.Text;
 form3.StatusBar1.Panels[3].text:='Selcall number: '+Form6.Edit3.Text;
 form1.Rx:=StrToFloat(Form6.Edit1.Text);
 form1.Tx:=StrToFloat(Form6.Edit2.Text);
 form1.selcall:=StrToInt(Form6.Edit3.Text);
 Form6.Hide;
end;

procedure TForm6.SpeedButton1Click(Sender: TObject);
begin
 Form6.Hide;
 Form8.Show;
end;

end.
```

9.6.7 Unit8.pas

```
unit Unit8;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, Menus;

type
  TForm8 = class(TForm)
    ListBox1: TListBox;
    ComboBox1: TComboBox;
    Label1: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    MainMenu1: TMainMenu;
    Plik1: TMenuItem;
    Otwrz1: TMenuItem;
    Zapisz1: TMenuItem;
    Edycja1: TMenuItem;
    Ususatacje1: TMenuItem;
    Edytujstacje1: TMenuItem;
    Dodajstacje1: TMenuItem;
    pom: TMemo;
    zmienazwe1: TMenuItem;
    zmienumerselcall1: TMenuItem;
    usuczestotliwo1: TMenuItem;
    dodajczstotliwo1: TMenuItem;
    procedure FormActivate(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
    procedure ListBox1DblClick(Sender: TObject);
    procedure ListBox1KeyPress(Sender: TObject; var Key: Char);
    procedure Button1Click(Sender: TObject);
    procedure Otwrz1Click(Sender: TObject);
    procedure Zapisz1Click(Sender: TObject);
    procedure Ususatacje1Click(Sender: TObject);
    procedure Dodajstacje1Click(Sender: TObject);
    procedure zmienazwe1Click(Sender: TObject);
    procedure zmienumerselcall1Click(Sender: TObject);
```

```
    procedure usuczestotliwo1Click(Sender: TObject);
    procedure dodajczstotliwo1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
    pos1, pos2: integer;
    procedure refresh_frequencies;
  end;

var
  Form8: TForm8;

implementation

uses Unit1, Unit3;

{$R *.DFM}
procedure TForm8.refresh_frequencies;
var
 i: integer;
 Rx_l, Tx_l, selcall_l,a:string;
begin
 //znajdujemy wybrana stacje w ksiazce
 for i:=0 to Form1.KsiazkaAdresowa.Lines.count-1 do begin
  if (Form1.KsiazkaAdresowa.Lines[i]=
     '#'+ListBox1.Items[ListBox1.ItemIndex]) then begin
   pos1:=i;
   break;
  end;
 end;
 //znajdujemy koniec sekcji tej stacji w ksiazce
 for i:=pos1+1 to Form1.KsiazkaAdresowa.Lines.count-1 do begin
  if (Form1.KsiazkaAdresowa.Lines[i][1]='#') then begin
   pos2:=i;
   break;
  end;
 end;
 selcall_l:=Form1.KsiazkaAdresowa.Lines[pos1+1];
 i:=pos1+2;
 ComboBox1.Items.Clear;
```

```
 ComboBox1.Text:='';
 while (i<pos2-1) do begin
  rx_l:=Form1.KsiazkaAdresowa.Lines[i];
  if length(rx_l)>7 then rx_l:='1000   ' else begin
   while (length(rx_l)<7) do begin
    rx_l:=rx_l+' ';
   end;
  end;
  tx_l:=Form1.KsiazkaAdresowa.Lines[i+1];
  if length(tx_l)>7 then tx_l:='1000   ' else begin
   while (length(tx_l)<7) do begin
    tx_l:=tx_l+' ';
   end;
  end;
  a:='Rx='+rx_l+'    '+'Tx='+tx_l+'  ';
  ComboBox1.Items.Add(a);
  inc(i,2);
 end;
 ComboBox1.ItemIndex:=1;
 Edit1.Text:=selcall_l;
end;

procedure TForm8.FormActivate(Sender: TObject);
 var
  i: integer;
begin
 //wypisujemy stacje
 ListBox1.Items.Clear;
 for i:=0 to Form1.KsiazkaAdresowa.Lines.count-1 do begin
  if (Form1.KsiazkaAdresowa.Lines[i][1]='#') then
     ListBox1.Items.Add(copy(Form1.KsiazkaAdresowa.Lines[i],
     2,length(Form1.KsiazkaAdresowa.Lines[i])-1));
 end;
 Listbox1.ItemIndex:=1;
 refresh_frequencies;
end;

procedure TForm8.ListBox1Click(Sender: TObject);
begin
 refresh_frequencies;
end;
```

```
procedure TForm8.ListBox1DblClick(Sender: TObject);
begin
 refresh_frequencies;
end;

procedure TForm8.ListBox1KeyPress(Sender: TObject; var Key: Char);
begin
 refresh_frequencies;
end;

procedure TForm8.Button1Click(Sender: TObject);
begin
 Form1.Selcall:=StrToInt(Edit1.Text);
 Form1.Rx:=StrToFloat(copy(ComboBox1.Text,4,7));
 Form1.Tx:=StrToFloat(copy(ComboBox1.Text,18,7));
 form3.StatusBar1.Panels[1].text:='Rx: '+FloatToStr(Form1.Rx);
 form3.StatusBar1.Panels[2].text:='Tx: '+FloatToStr(Form1.Tx);
 form3.StatusBar1.Panels[3].text:='Selcall number: '+IntToStr(Form1.Selcall);
 Form8.hide;

end;

procedure TForm8.Otwrz1Click(Sender: TObject);
begin
 if Form1.OpenDialog1.Execute then begin
  Form1.KsiazkaAdresowa.Lines.LoadFromFile(Form1.OpenDialog1.FileName);
 end;
 FormActivate(self);
end;

procedure TForm8.Zapisz1Click(Sender: TObject);
begin
 if Form1.SaveDialog1.Execute then begin
  Form1.KsiazkaAdresowa.Lines.SaveToFile(Form1.SaveDialog1.FileName);
 end;
end;

procedure TForm8.Usustacje1Click(Sender: TObject);
var
 i: integer;
```

```
begin
 pom.text:='';
 for i:=0 to pos1-1 do begin
  pom.lines.add(Form1.KsiazkaAdresowa.Lines[i]);
 end;
 if pos2<Form1.KsiazkaAdresowa.Lines.Count then begin
  for i:=pos2 to Form1.KsiazkaAdresowa.Lines.Count-1 do begin
   pom.lines.add(Form1.KsiazkaAdresowa.lines[i]);
  end;
 end;
 Form1.KsiazkaAdresowa.text:=pom.Text;
 FormActivate(self);
end;

procedure TForm8.Dodajstacje1Click(Sender: TObject);
var
 i, pos: integer;
begin
 pos:=Form1.KsiazkaAdresowa.Lines.count-1;
 for i:=0 to Form1.KsiazkaAdresowa.Lines.count-1 do begin
  if Form1.KsiazkaAdresowa.Lines[i]='#' then pos:=i;
 end;
 Form1.KsiazkaAdresowa.Lines[pos]:='#nowa stacja';
 Form1.KsiazkaAdresowa.Lines.Add('8888');
 Form1.KsiazkaAdresowa.Lines.Add('#');
 FormActivate(self);
end;

procedure TForm8.zmienazwe1Click(Sender: TObject);
begin
 Form1.KsiazkaAdresowa.lines[pos1]:='#'+InputBox
      ('Telex','Podaj nową nazwe stacji','');
 FormActivate(self);
end;

procedure TForm8.zmienumerselcall1Click(Sender: TObject);
begin
 Form1.KsiazkaAdresowa.lines[pos1+1]:=InputBox
      ('Telex','Podaj nowy numer selcall stacji','');
 FormActivate(self);
end;
```

```
procedure TForm8.usuczestotliwo1Click(Sender: TObject);
var
 i: integer;
begin
 pom.text:='';
 for i:=0 to pos1+(ComboBox1.ItemIndex*2)+1 do begin
  pom.lines.add(Form1.KsiazkaAdresowa.Lines[i]);
 end;
 if pos1+(ComboBox1.ItemIndex*2)+3<
        Form1.KsiazkaAdresowa.Lines.count-1 then begin
  for i:=pos1+(ComboBox1.ItemIndex*2)+4 to
        Form1.KsiazkaAdresowa.Lines.count-1 do begin
   pom.lines.add(Form1.KsiazkaAdresowa.Lines[i]);
  end;
 end;
 Form1.KsiazkaAdresowa.Text:=pom.text;
 FormActivate(self);
end;

procedure TForm8.dodajczstotliwo1Click(Sender: TObject);
var
 i: integer;
begin
 pom.text:='';
 for i:=0 to pos1+1 do begin
  pom.lines.add(Form1.KsiazkaAdresowa.Lines[i]);
 end;
 pom.lines.add(InputBox('Telex','Rx = ?',''));
 pom.lines.add(InputBox('Telex','Tx = ?',''));
 if pos1+1<Form1.KsiazkaAdresowa.Lines.count-1 then begin
  for i:=pos1+2 to Form1.KsiazkaAdresowa.Lines.count-1 do begin
   pom.lines.add(Form1.KsiazkaAdresowa.Lines[i]);
  end;
 end;
 Form1.KsiazkaAdresowa.Text:=pom.text;
 FormActivate(self);
end;

end.
```