



Politechnika Wrocławska

Sprawozdanie 7

Ćwiczenie 7. Modemy. Transmisja sygnałów cyfrowych

Krzysztof Zalewa, Wiktor Wojnar

2.2.2025

Spis treści

1	Wstęp teoretyczny	2
1.1	Metody konfiguracji modemów.	2
1.1.1	Komendy Hayes	2
2	Zadanie laboratoryjne	3
2.1	Treść zadania	3
2.2	Opis działania programu	3
2.3	Kod programu	3
3	Źródła	6

1 Wstęp teoretyczny

1.1 Metody konfiguracji modemów.

1.1.1 Komendy Hayes

Komendy Hayes (zbiór komend AT) to specjalny język komend oryginalnie stworzony na potrzeby modemu firmy Hayes w 1981. Obecnie zbiór ten stał się standardem i jest używany w większości nowoczesnych urządzeń. Zbiór ten można podzielić na cztery grupy:

1. **Komendy podstawowe** - Duża litera i cyfra.Np I0.
2. **Komendy rozszerzone** - Znak & ,duża litera i cyfra. Rozszerza podstawowy zbiór więc I0 != &I0.
3. **Komendy własne** - Zwykle poprzedzone \lub %. Te komendy są bardzo różne ponieważ są pozostawione potrzebom producentów.
4. **Komendy rejestrów** - S n gdzie n to numer rejestru. Bezpośrednio modyfikuje miejsce w pamięci urządzenia.

Modem A	Modem B	Komentarz
ATDT12345678987		Użytkownik A podaje komendę do modemu A AT-tention; D-Dial; T-Touch-Tone; Zadzwoń na ten numer 12345678987
	Dzwoni	Modem A rozpoczyna dzwonenie na modem B. Modem B daje znać o przychodzącym połączeniu
	ATA	Komputer B odbiera połączenie
Połączenie	Połączenie	Oba modemy wyświetlają informację o poprawnym połączeniu.
qwerty	qwerty	Kiedy modemy są połączone
	+++	Przejsie do trybu poleceń
	OK	Potwierdzenie wykonania polecenia
	ATH	Komenda przzerwania polecenia
Brak połączenia	OK	Potwierdzenie wykonania polecenia

Element	Opis
AT	Przedrostek komendy- poprzedza wiersz komend.
<CR>	Znak powrotu karetki (znak nowego wiersza)-przerwano wiersz komend.
A	Idź na hak, pozostaj w trybie poleceń.
A/	Powtórz poprzedni wiersz komend. Ta komenda nie jest poprzedzona adresem AT lub po niej następuje < CR>/.
B0	Wybierz standard CCITT V.22 dla komunikacji 1200 b/s.
B1	Wybierz standard Bell 212A dla komunikacji 1200 b/s.
D	Wprowadź tryb pochodzenia, wybierz numer, który następuje, a następnie spróbuj przejść do trybu z połączeniem. D jest zazwyczaj stosowana przez T dla tonu, P dla impulsu może być również używany.
DS=n	Wybierz numer zapisany w lokalizacji n
E0	Wyłącz echo znakowe w stanie komendy.
E1	Włącz echo znakowe w stanie komendy.
H0	Idź na-hak (rozłącz się przez telefon).
H1	Obsługiwać wyłącznik hakowy i przekaźnik pomocniczy.
I0	Zwraca kod identyfikacyjny produktu.
I1	Wykonaj sumę kontrolną oprogramowania wbudowanego ROM; zwróć sumę kontrolną.
I2	Wykonaj sumę kontrolną w pamięci ROM oprogramowania wbudowanego; w wyniku tego zwracane są wartości OK lub ERROR .
L0	Wyłącz głośnik.
L1	Niska objętość głośnika.
L2	Głośnik średniego głośnika.

Rysunek 1: Przykładowe komendy Hayes'a

2 Zadanie laboratoryjne

2.1 Treść zadania

W ramach zajęć laboratoryjnych należało skonfigurować i przetestować połączenie między dwoma modemami. Następnie należało napisać program który będzie w stanie wykonywać funkcje takie jak łączenie się z modemem, transmisja terminal-terminal oraz przesył plików.

2.2 Opis działania programu

Program może wykonywać jednocześnie role nadawcy i odbiorcy(Po wybraniu odpowiedniej opcji). Po wybraniu opcji nadawania program łączy się z wybranym modemem (Modem docelowy powinien być w trybie auto odbierania). Następnie można wybrać opcje przesyłu plików lub tekstu. Po wybraniu trybu odbierania program oczekuje na połączenie. Kiedy połączenie zostanie potwierdzone odbierane i wyświetlane są wiadomości z modemu 1.

2.3 Kod programu

```

1 import serial
2 import time
3 from serial import EIGHTBITS, PARITY_NONE, STOPBITS_ONE
4 from xmodem import XMODEM
5
6
7 def connect_to_modem(port, baudrate):
8     try:
9         # Otwieramy połączenie szeregowe z modemem

```

```

10     ser = serial.Serial(port, baudrate, bytesize=EIGHTBITS, parity=PARITY_NONE,
    ↪ stopbits=STOPBITS_ONE, timeout=1)
11     print(f"Połączono z modemem na porcie {port}")
12     return ser
13 except serial.SerialException as e:
14     print(f"Nie udało się połączyć z modemem: {e}")
15     return None
16
17 def dial_number(ser, number):
18     # Wysyłamy polecenie ATD w celu nawiązania połączenia telefonicznego
19     at_command = f"ATD{number}\r"
20     ser.write(at_command.encode('utf-8'))
21     print(f"Wysłano polecenie ATD: {number}")
22
23     # Oczekiwanie na odpowiedź
24     time.sleep(30)
25     response = ser.read_all().decode()
26
27     if "CONNECT" in response:
28         print("Połączono z modemem!")
29         return True
30     else:
31         print("Nie udało się połączyć.")
32         return False
33
34
35 def answer_call(ser):
36     print("Oczekiwanie na połączenie przychodzące...")
37     try:
38         while True:
39             if ser.in_waiting > 0:
40                 response = ser.read_all().decode()
41                 print(f"Odebrano: {response}")
42
43                 # Sprawdzamy, czy modem odbiera sygnał dzwonienia
44                 if "RING" in response:
45                     print("Wykryto połączenie przychodzące.")
46                     ser.write("ATA\r".encode()) # Wysyłamy komendę ATA, aby
    ↪ odpowiedzieć
47                     time.sleep(5) # Czekamy na odpowiedź
48
49                     response = ser.read_all().decode()
50                     print(f"Odpowiedź po ATA: {response}")
51
52                     if "CONNECT" in response:
53                         print("Połączenie zostało nawiązane!")
54                         return True
55                     else:
56                         print("Nie udało się połączyć.")
57                         return False
58 except KeyboardInterrupt:
59     print("Przerwano oczekiwanie na połączenie.")
60     return False
61
62
63 def terminal_to_terminal(ser):

```

```

64     print("Rozpaczynam sesję terminalową. Wpisz 'exit', aby zakończyć.")
65     try:
66         while True:
67             # Odczyt danych
68             if ser.in_waiting > 0:
69                 data = ser.read(ser.in_waiting).decode()
70                 print(f"Modem: {data}")
71
72             # Wprowadzenie danych od użytkownika
73             user_input = input("Ty: ")
74             if user_input.lower() == "exit":
75                 break
76             ser.write((user_input + "\r\n").encode(encoding='utf-8')) # Wysyłamy
77                                     ↪ dane
78         except KeyboardInterrupt:
79             print("Sesja przerwana.")
80
81     def send_file_via_xmodem(ser, filename):
82         def getc(size, timeout=1):
83             return ser.read(size) or None
84
85         def putc(data, timeout=1):
86             return ser.write(data)
87
88         modem = XMODEM(getc, putc)
89
90         # Otwieramy plik do wysłania
91         with open(filename, 'rb') as f:
92             print(f"Rozpaczynam przesyłanie pliku {filename}...")
93             status = modem.send(f)
94             if status:
95                 print("Plik został pomyślnie wysłany!")
96             else:
97                 print("Wystąpił błąd podczas przesyłania pliku.")
98
99
100     # Funkcja główna
101     def main():
102         port = input("Podaj port szeregowy: ")
103         baudrate = 9600 # Szybkość transmisji
104         modem_number = "3965" # Numer modemu docelowego
105
106         ser = connect_to_modem('COM1', baudrate)
107         if not ser:
108             return
109
110         print("Wybierz tryb:")
111         print("1: Wykonaj połączenie")
112         print("2: Odbierz połączenie")
113         choice = input("Twój wybór: ")
114
115         if choice == "1":
116             if dial_number(ser, modem_number):
117                 print("Wybierz tryb:")
118                 print("1: Konwersacja terminal-terminal")

```

```

119         print("2: Transmisja pliku (XModem)")
120         sub_choice = input("Twój wybór: ")
121
122         if sub_choice == "1":
123             terminal_to_terminal(ser)
124         elif sub_choice == "2":
125             filename = input("Podaj nazwę pliku do wysłania: ")
126             send_file_via_xmodem(ser, filename)
127         else:
128             print("Nieprawidłowy wybór.")
129     elif choice == "2":
130         if answer_call(ser):
131             print("Połączenie zostało nawiązane. Wybierz tryb:")
132             print("1: Konwersacja terminal-terminal")
133             print("2: Transmisja pliku (XModem)")
134             sub_choice = input("Twój wybór: ")
135
136             if sub_choice == "1":
137                 terminal_to_terminal(ser)
138             elif sub_choice == "2":
139                 filename = input("Podaj nazwę pliku do wysłania: ")
140                 send_file_via_xmodem(ser, filename)
141             else:
142                 print("Nieprawidłowy wybór.")
143         else:
144             print("Odbieranie połączenia nie powiodło się.")
145     else:
146         print("Nieprawidłowy wybór.")
147
148     # Zamykamy połączenie
149     ser.close()
150     print("Połączenie zostało zakończone.")
151
152
153 if __name__ == "__main__":
154     main()
155
156

```

Fragment kodu 1: Fragment kodu z programu

3 Źródła

1. https://en.wikipedia.org/wiki/Hayes_AT_command_set
2. <https://www.ibm.com/docs/pl/aix/7.3?topic=troubleshooting-commands>