

Programowa obsługa telefonów GSM firmy Nokia

W EP 8/2002 szczegółowo opisaliśmy polecenia AT oraz sposób programowania modułów GSM.

Teraz skupiamy się na ekspresowym przybliżeniu sposobu nawiązywania komunikacji z telefonami GSM firmy Nokia, która - nie tylko w naszym kraju - oferuje niezwykle wiele popularnych modeli telefonów.



Nokia wyprodukowała ogromną liczbę modeli telefonów komórkowych. Niektóre z nich są bardzo udane i cieszą się ogromną popularnością również i dzisiaj. Takimi przykładami są np. nieśmiertelne modele: 5110, 3210, 6110. Na rynku ciągle pojawiają się nowe modele tej firmy. Ważną zaletą jest możliwość sterowania tych telefonów tym samym protokołem. Często pojawiające się zapytania na grupie dyskusyjnej pl.misc.elektronika skłoniły mnie do opisanie protokołu sterującego tymi telefonami. Pytania często dotyczyły możliwości wysyłania, odbierania SMS (krótkich wiadomości tekstowych) za pomocą układów mikroprocesorowych, czy sterowników. Wielu z nas chciałoby wykorzystać do swoich amatorskich układów telefon komórkowy, by móc zdalnie nim sterować lub być informowanym o zdarzeniach ze swojego układu mikroprocesorowego. Do niedawna podłączanie i komunikacja z telefonem komórkowym za pomocą kabelek była domeną tylko dla nielicznych wtajemniczonych guru. Natomiast proste w sterowaniu za pomocą komend AT moduły GSM były (i są) drogie i praktycznie niedostępne dla przeciętnego elektronika - amatora.

Obecnie firmy produkujące telefony komórkowe coraz częściej implementują w swoich aparatach „kawałek” oprogramowania pozwalający sterować nimi za pomocą komend AT. Opisane wcześniej nieśmiertelne 5110 (i podobne) mają jedną ważną zaletę - można wejść w ich posiadanie za przysłowiowe piwo lub po prostu wyciągnąć z szuflady, gdzie leżą niewykorzystane. W Internecie można znaleźć mnóstwo schematów in-

terfejsów służących do podłączenia telefonu komórkowego do PC, czy mikrokontrolera. Zajmijmy się zatem opisem protokołu.

Ramki protokołu Nokia 5110, 3210, 6110

Transmisja danych z telefonami Nokia odbywa się za pomocą specjalnego protokołu składającego się z ramek binarnych. Protokół ten nazywa się w skrócie FBUS.

Opisany przeze mnie protokół pozwala na komunikację i dostęp do większości funkcji następujących telefonów firmy Nokia: 3210, 3310, 8210/8850,

- Telefon komórkowy odpowiada ramką potwierdzającą (pole określające typ ramki 7Fh).
- Telefon odpowiada ramką z porcją danych.

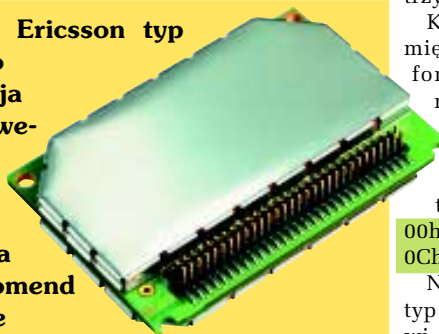
Przed pierwszą transmisją musimy przesłać ciąg o wartości 55h i długości około 230 bajtów (Typowe wartości to 231, 233, 246, 250, 253). Transmisja ta ma na celu zsynchronizowanie układu odbiorczego telefonu. Pole odpowiedzialne za typ ramki w ramce danych jest identyczne jak w ramce żądającej tych danych. Otrzymaną ramkę musimy potwierdzić w określonym czasie, w przeciwnym wypadku telefon komórkowy powtórzy ramkę jeszcze trzy razy i zakończy transmisję.

Każda ramka przesyłana pomiędzy mikrokontrolerem a telefonem komórkowym NOKIA rozpoczyna się preambułą - polu o wartości 1Eh. Po preambule następują dwa bajty określające kierunek transmisji:

| | | |
|-----|-----|-------------|
| 00h | 0Ch | do telefonu |
| 0Ch | 00h | z telefonu |

Następny jest bajt określający typ ramki danych. Bajt 7Fh mówi nam, że ramka jest ramką potwierdzającą. Następnym w kolej-

Moduł GSM firmy Ericsson typ GM47/48. Jest to przemysłowa wersja telefonu komórkowego, pozbawiona klawiatury i wyświetlacza, sterowanie odbywa się za pomocą komend AT poprzez złącze RS232.



61xx/51xx, 62xx, 7110, 3330, 3110/8110. Domyślne parametry transmisji protokołu są następujące: 115200 bps, 8 bitów danych, 1 bit stopu, bez parzystości. Niestety prędkość transmisji jest dość wysoka i wiele popularnych mikrokontrolerów jednocukładowych z rodziny '51 nie potrafi sobie z nią poradzić. Komunikacja z telefonem odbywa się według poniższego schematu:

- Wysyłamy ramkę z zapytaniem do telefonu komórkowego.



Rys. 1



Rys. 2

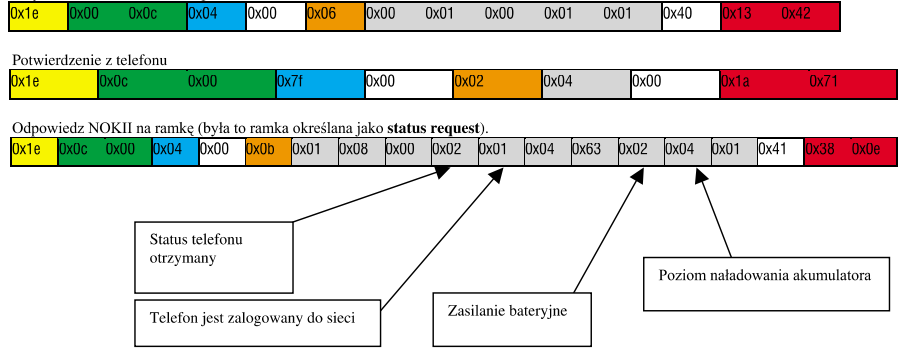
ności jest bajt zerowy. Po bajcie zerowym następuje pole określające długość danych w ramce, tzw. *payload length* - gdy długość jest nieparzysta na koniec danych dodajemy jeden bajt zerowy. Po bajcie długości, jeżeli nie jest to ramka potwierdzająca, umieszczamy bajt zerowy. Ramka kończy się licznikiem ramek oraz dwoma bajtami sumy kontrolnej. Gdy długość ramki jest nieparzysta przed dwoma bajtami sumy kontrolnej pojawia się bajt o wartości 00h. Pierwszy z bajtów sumy kontrolnej utworzony jest za pomocą funkcji XOR z wszystkich nieparzystych bajtów. Drugi bajt sumy kontrolnej jest utworzony za pomocą funkcji XOR z wszystkich parzystych bajtów. Ponieważ XOR odbywa się dla tej samej ilości bajtów (parzystych i nieparzystych) - stąd powód istnienia dodatkowego pola w przypadku nieparzystej długości. Licznik ramek to bajt składający się z wartości 40h (starszy półbajt), oraz cyklicznie zmieniającej się liczby od 0 do 7 (młodszy półbajt). Pole to jest zwiększane o jeden po wysłaniu ramki, nie jest natomiast zwiększane po retransmisji ramek na skutek błędów. W ramach potwierdzających bajt - licznik ramek składa się tylko z numeru (odpowiada on numerowi z potwierdzanej ramki). Bardziej znaczący półbajt jest zerem.

Wysyłanie SMS-ów

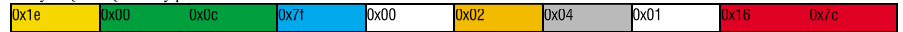
Aby zrozumieć strukturę ramki i sposób wysyłania SMS za pomocą telefonu NOKIA najlepiej skorzystać z napisanego przeze mnie programu. Jego główne okno pokazano na rys. 1. Dla podanego centrum SMS, numeru telefonu odbiorcy, oraz treści SMS-a (pokazane na rys. 2) program generuje poniższą ramkę:

```
[0x1e]-----Preambuła
[0x00][0x0c]----Ramka do telefonu
[0x02]-----Typ ramki
[0x00]
[0x61]-----Długość ramki
[0x00][0x01][0x00]---Nagłówek ramki
[0x01][0x02][0x00]
-----Centrum SMS-----
[0x07]-----Długość numeru
[0x91]-----Typ Numeru numeru
-----Spakowany numer
[0x84][0x06][0x04][0x01][0x01][0xf1][0x00][0x00][0x00][0x00]
-----
[0x11]-----Format pola ważności SMSa
[0x00]-----Message Reference
[0x00]-----Identyfikator protokołu
```

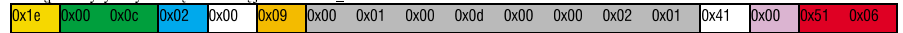
Przykład – ramka z rozkazem przesłania statusu telefonu.



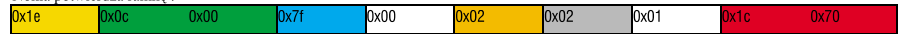
Powyższą ramkę musimy potwierdzić.



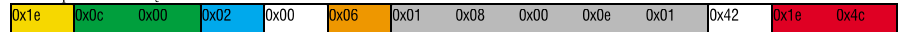
Następnie wysyłamy ramkę określoną jako **connect_1**.



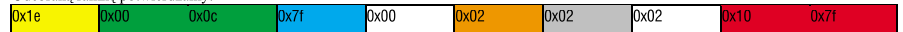
Nokia potwierdza ramkę:



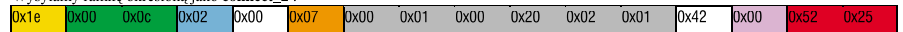
Oraz odpowiada ramką:



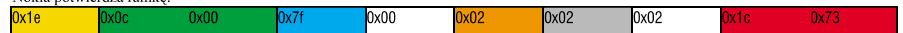
Odebraną ramkę potwierdzamy:



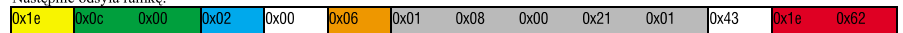
Wysyłamy ramkę określoną jako **connect_2**:



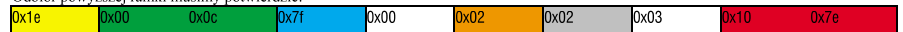
Nokia potwierdza ramkę:



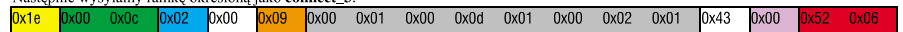
Następnie odsyła ramkę:



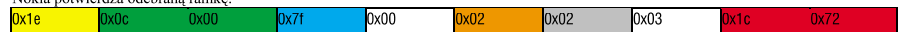
Odbiór powyższej ramki musimy potwierdzić:



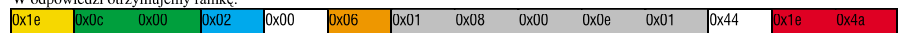
Następnie wysyłamy ramkę określoną jako **connect_3**:



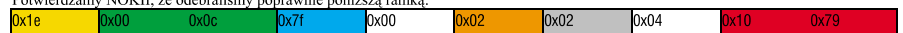
Nokia potwierdza odebraną ramkę:



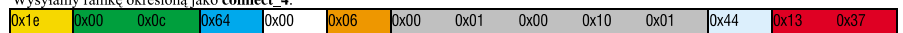
W odpowiedzi otrzymujemy ramkę:



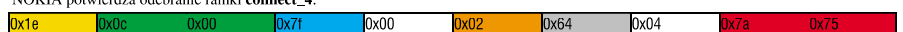
Potwierdzamy NOKII, że odebraliśmy poprawnie poniższą ramkę:



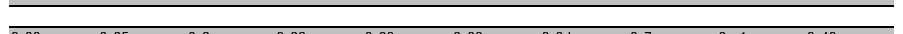
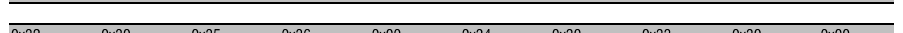
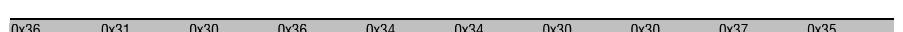
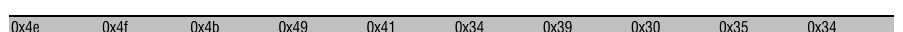
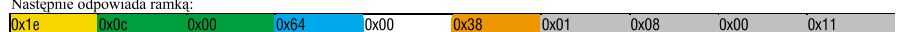
Wysyłamy ramkę określoną jako **connect_4**:



NOKIA potwierdza odebranie ramki **connect_4**:



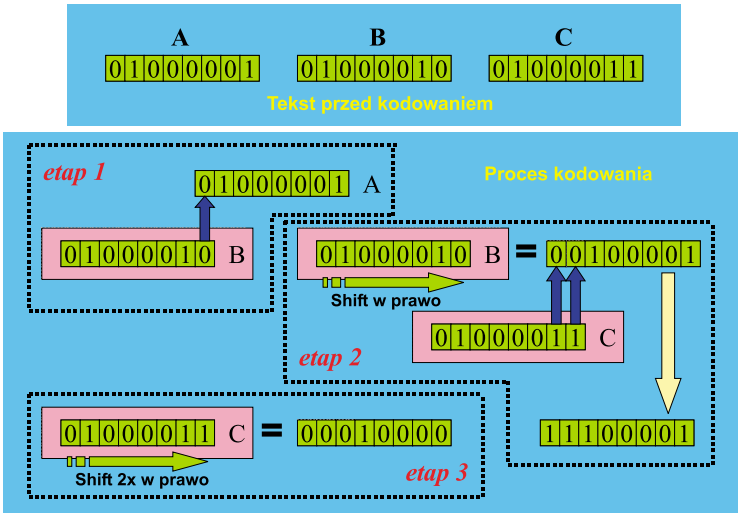
Następnie odpowiada ramką:



Mikrokontroler potwierdza odebranie powyższej ramki poprzez wysłanie:



W odpowiedzi na ramkę **connect_4** telefon odsyła m.in. IMEI, model i *revision code*. Po pomyślnej wymianie wstępnych danych, na wyświetlaczu telefonu komórkowego powinna wyświetlić się ikona „Akcesoria Nokii”.



Rys. 3

```
[0xf0]-----Schemat kodowania SMSa (7, 8 bit)
[0x3c]-----Liczba znaków w SMSie

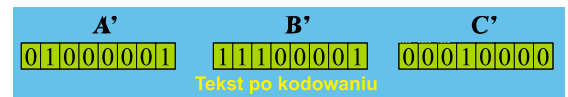
-----Telefon adresata-----
[0x0b]-----Długość numeru
[0x91]-----Typ Numeru numeru
-----Spakowany numer
[0x84][0xd6][0x20][0x00][0x03][0xf0][0x00][0x00][0x00][0x00]
-----
[0x32]-----Ważność SMS-a
-----Sześć bajtów zerowych
[0x00]
[0x00]
[0x00]
[0x00]
[0x00]
[0x00]
-----Spakowana tresc SMSa-----
[0xc4][0x37][0xd8][0x5d][0xc9e][0xc3][0xd1][0xc3][0xe6][0xc14][0xd44]
[0x2e][0xcfe][0xe9][0xf7b][0xcde][0xd5][0x9a][0xc2][0xcdb][0x63]
[0x75][0x98][0xe3][0x4e][0x97][0x41][0x64][0x76][0x18][0x54][0x64]
[0x97][0x37][0x74][0xf9][0xcdb][0x9d][0x5e][0xa7][0xd1][0x50][0x79]
[0x78][0x4d][0xcf][0x8f][0xf5][0xee][0xb2][0xda][0x05]
```

```
[0x01]
[0x43]-----Numer sekwencji
[0x00]-----Bajt dodatkowy
[0x77][0x11]-----Dwa bajty CRC
```

Objaśnienia dotyczące pól ramki znajdują się w tab. 1.

Kodowanie znaków w SMS odbywa się w trybie siedmiobitowym. Każdy znak tekstu wykorzystującego standardowe znaki alfabetu łańciskiego, czyli tzw. *plain text*, daje się opisać za pomocą 7 bitów. Przed „włożeniem” tekstu do 8-bitowej ramki odbywa się proces pakowania według schematu pokazanego na rys. 3, dzięki któremu uzyskujemy kompresję liczby przesyłanych danych o wartości nieco ponad 14%.

Każdą literę alfabetu łańciskiego (plus niektóre znaki) możemy zapisać wykorzystując do tego celu siedem bitów. Bit ósmy pozostaje niewykorzystany - w jego



Rys. 4

miejsce umieszczamy najmniej znaczący bit z drugiego znaku (*etap 1*). Drugi znak przesuwamy o jeden bit w lewo. Mamy teraz dwa bity wolne na najbardziej znaczącej pozycji znaku drugiego. Umieszczamy tam dwa bity z najmniej znaczącej pozycji znaku trzeciego (*etap 2*).

Trzeci znak przesuwamy o dwa bity w prawo (o dwa, ponieważ dwa bity przenieśliśmy do znaku drugiego). W ten sposób mamy trzy bity wolne na najbardziej znaczącej pozycji (*etap 3*). I tak dalej. Zakodowane dane pokazano na rys. 4.

Dzwonienie

Za pomocą prezentowanego wcześniej programu można także przygotować ramkę danych umożliwiającą wybranie numeru telefonu abonenta i ustalić połączenie foniczne. W tym celu w głównym oknie programu (rys. 1) należy wybrać *Zadzwoń* i w wyświetlonym oknie (rys. 5) podać numer abonenta. Program generuje ramkę jak poniżej:

```
[0x1e]-----Preambuła
[0x00][0x0c]---Ramka do telefonu
[0x01]-----Typ ramki
[0x00]
[0x15]-----Długość ramki
[0x00][0x01][0x00]---Nagłówek ramki
[0x01]
[0x06]-----Długość numeru telefonu

-----Numer telefonu-----
[0x31][0x32][0x31][0x32][0x31][0x32]
-----
[0x05][0x01][0x01][0x05][0x81][0x01][0x00][0x00][0x01]
[0x40]-----Numer sekwencji
[0x00]-----Bajt dodatkowy
[0x63][0xa1]---Dwa bajty CRC
```

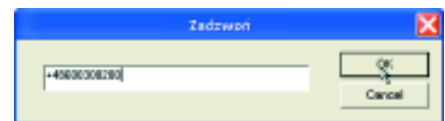
Długość numeru telefonu - liczba znaków (cyfr) numeru telefonu abonenta do którego chcemy się zadzwonić

Telefon - numer telefonu jest inaczej kodowany niż w przypadku SMS-a. Tutaj każdy bajt ramki, to kolejny kod ASCII cyfry z numeru telefonu.

Paweł Dienwebel

Literatura

- [1]. Strona Marcina Wiacka - <http://marcin-wiacek.fkn.pl>
- [2]. <http://www.gnokii.org>



Rys. 5

| Tab. 1. Objasnienia dotyczące pól ramki | |
|---|---|
| Długość ramki | Obliczana jest od pola za polem długości aż do dwóch bajtów CRC (CRC nie są wliczane do długości ramki) |
| Pole określające numer SMS | Długość w znakach (liczba cyfr) |
| | Typ numeru: 0x91 - międzynarodowy (w formacie +486.....) 0x81 - nieznanym |
| Pole ważności | 1 godzina 0x0b 6 godzin 0x47 24 godziny 0xa7 3 dni 0xa9 tydzień 0xad maksymalny czas 0xff |
| Message Reference | SMS tekstowy 0x00 Faks 0x22 Wiadomość głosowa 0x24 Wiadomość w formacie ERMES 0x25 Pager 0x26 Email w formacie NOKIA 8110(UCI) 0x2d Email 0x32 Wiadomość w formacie X400 0x31 |
| Schemat kodowania SMS | W SMS dane są siedmiobitowe. Ten tryb jest używany do przesyłania wiadomości tekstowych. Pole <i>Message Reference</i> ma wtedy wartość 0x00. Kodowanie 8 bit dla pozostałych wartości <i>Message Reference</i> . |
| Numer sekwencji | Składa się z 0x40 (starszy półbajt), oraz liczby 0..7 (młodszy półbajt). Młodszy półbajt jest każdorazowo inkrementowany po przesłaniu ramki. W przypadku retransmisji wartość ta nie jest zwiększana. |

Dodatkowe informacje

Oprogramowanie prezentowane w artykule wraz z kodami źródłowymi opublikujemy na CD-EP10/2002B, jest ono dostępne także na naszej stronie w dziale *Download>Dokumentacje*.