

# Centrala alarmowa z powiadomieniem GSM, część 2

## AVT-526



W drugiej części artykułu przedstawiamy tajniki przygotowania programu sterującego pracą centrali ze zdalnym powiadamianiem za pomocą telefonu GSM. Jak się okazuje, wysyłanie SMS-ów nie jest aż tak trudne jak się powszechnie uważa.

**Rekomendacje:** nowoczesny i bardzo skuteczny system zdalnej ochrony mienia, dzięki któremu informacja o włamaniu dotrze do Ciebie szybko i w dowolne miejsce.

### Wysyłanie SMS

„Jak wysłać wiadomości SMS za pomocą mikrokontrolera?“, to pytanie bardzo często występujące w listach kierowanych do Redakcji oraz zadawane na forum grup dyskusyjnych w Internecie, dlatego też poświęcę mu nieco więcej miejsca. Obydwa z posiadanych przez mnie modemów GSM obsługują dwa tryby przesyłania wiadomości SMS. Są one oznaczone numerami 0 i 1. Odpowiedź telefonu lub modemu na rozkaz „AT+CMGF?“ lub „AT+CMGF=?“ będzie zawierać listę obsługiwanych trybów pracy. Ja dla prezentowanej aplikacji wybrałem tryb 1, moim zdaniem najodpowiedniejszy. Nawiasem mówiąc, wolę nie wyrażać swojej opinii na temat kodowania wiadomości w trybie 0, zwanym również *PDU mode*...

Podane poniżej informacje opracowane zostały na podstawie dokumentacji firmy Siemens dla modemu GSM M20. Zainteresowanym poszerzeniem swojej wiedzy na ten temat odsyłam szczególnie do czterech źródeł:

1. Dokumentacji modemu M20 Siemens dostępnej na stronie pro-

ducenta pod nazwą „sms\_guid\_v1\_0.doc“. Autorem jest T. Schuh.

2. Noty aplikacyjnej umieszczonej pod adresem [http://www.mcselec.com/an\\_31.htm](http://www.mcselec.com/an_31.htm). Autorem jest Vilko Sustić.

3. Noty aplikacyjnej umieszczonej pod adresem [http://www.mcselec.com/an\\_117.htm](http://www.mcselec.com/an_117.htm). Autorem jest Warren Read.

4. Informacji zawartych na stronie internetowej <http://www.dreamfabric.com/sms/>.

### Mode 0 (PDU)

W tym trybie na projektanta aplikacji spada obowiązek zakodowania całej informacji dotyczącej wysłanego komunikatu. Przeanalizujmy przykład (rys. 3) zaczerpnięty z materiałów firmy Siemens, ilustrujący przesyłanie wiadomości o treści „THE BIG BROWN FOX“ pod numer telefonu +991234567.

Wyjaśnijmy znaczenie poszczególnych pól komunikatu SMS:

1. *Długość pola SCA* jest opcjonalna dla niektórych telefonów. Jest to podana szesnastkowo liczba oktetów numeru Centrum Usług. Większość telefonów umożliwia wprowadzenie w tym

miejscu wartości 0x00, więc nie trzeba podawać i tym samym znać tego parametru.

2. *Typ numeru Centrum Usług* - podanie wartości 91 oznacza numerację międzynarodową (np. z prefiksem +48 dla Polski), 81 oznacza numerację lokalną, obowiązującą na terenie danego kraju.

3. *Numer Centrum Usług* to numer podany przez operatora, dzięki któremu możliwe jest przesyłanie wiadomości SMS. Np. dla sieci Plus GSM jest to numer +48601000100. Zwróćmy uwagę na sposób kodowania tego numeru. Na przykład numer „601000100“ zakodowany będzie w sposób następujący:

- numeracja krajowa, bez numeru kierunkowego: 81,
- nieparzysta liczba znaków - na końcu dodawana jest litera F, więc numer będzie miał postać „601000100F“,
- numer po kodowaniu będzie następujący: „06010010F0“.

4. Pierwszy oktet komunikatu SMS

Numer bitu	7	6	5	4	3	2	1	0
Nazwa bitu	TP- RP	TP- UDHI	TP- SRR	TP- VPF	TP- VPF	TP- RD	TP- MTI	TP- MTI

Znaczenie poszczególnych bitów jest następujące:

- **TP-RP** - ścieżka (numer) nadawcy; ustawienie bitu oznacza, że ścieżka dla odpowiedzi na wysłany SMS jest ustawiona,
- **TP-UDHI** - ustawienie bitu oznacza, że przesyłane dane użytkownika rozpoczynają się od nagłówka identyfikującego,
- **TP-SRR** - ustawienie tego bitu oznacza, że nadawca żąda raportu na temat odbioru wiadomości SMS przez odbiorcę,
- **TP-VPF** - bity określające format okresu ważności wiadomości SMS:

- **0 0**: TP-VPP nie występuje,
- **1 0**: TP-VPP obecne, format relacyjny (0...143 = (wartość parametru + 1) x 5 minut, 144...167 = 12 godzin + (wartość parametru - 143) x 30 minut, 168...196 = (wartość parametru - 166) x 1 dzień, 197...255 = (wartość parametru - 192) x 7 dni),

- **0 1**: format rozszerzony, 7 oktetów,
- **1 1**: format absolutny, 7 oktetów,

- **TP-RD** - ustawienie bitu powoduje, że duplikaty wiadomości są odrzucane,
- **TP-MTI** - typ komunikatu: wartość bitów 0 1 powoduje, że wiadomość przesyłana jest w trybie SMS-SUBMIT.

5. Pole statusu określające numer odniesienia dla wysyłanego komunikatu. Wpisanie wartości 0x00 powoduje, że numer telefonu, z którego wysyłamy SMS jest równocześnie numerem odniesienia.

6. Długość numeru odbiorcy SMS, podana w liczbie bajtów numeru.

7. Typ numeru odbiorcy SMS (81 lub 91).

8. Numer odbiorcy wiadomości SMS (kodowany w ten sam sposób, co numer Centrum Usług).

9. Identyfikator protokołu komunikacyjnego. Wartością typową, używaną przez większość sieci telefonów komórkowych jest 0x00.

10. Schemat kodowania danych. Wprowadzona wartość 0x00 oznacza, że znaki kodowane są w postaci 7-bitowych znaków ASCII.

11. Okres ważności wiadomości SMS. Liczba podana i interpretowana zgodnie z kodem reguły podanym przez bity TP-VPF.

12. Liczba oktetów bajtów tekstu.

13. Tekst wiadomości.

Zgodnie ze specyfikacją organizacji ETSI ([www.etsi.org](http://www.etsi.org)), wiadomość SMS może mieć do 160 znaków długości, przy czym każdy znak zakodowany jest w postaci słowa 7-bitowego. Znaki kodowane 8-bitowo zazwyczaj nie są przez telefony wyświetlane, natomiast są czasami używane jako sterujące (przesyłanie obrazów, dzwonek itp.). Komunikaty zakodowane w postaci liczb 16-bitowych (maksymalnie 70 znaków) są używane do przesyłania komunikatów zakodowanych według standardu Unicode (UCS2) i mogą być wyświetlane przez większość aparatów GSM. Niektóre z telefonów wyświetlają te komunikaty jako Flash SMS. My zajmujemy się kodowaniem standardowym, gdy kody znaków mają długość 7 bitów.

Prześledźmy sposób kodowania tekstu na przykładzie: zakodujemy wiadomość „ELEKTRONIKA“.

Tekst „ELEKTRONIKA“ zawiera 11 znaków nazywanych septetami ze względu na swą długość - 7 bitów. My musimy zmienić sposób kodowania z septetów na oktety - znaki o długości 8 bitów (**tab. 2**).

Pierwszy septet (E) zamieniany jest na oktet poprzez „zabranie“ najmniej znaczącego bitu następnego w kolejności znaku i wprowadzenie go na pozycję bitu 7 (0 + 1000101 = 01000101 tj. 0x45). Po tej operacji bit jest usuwany, więc następne przekształcenie (dla litery L) zabierze dwa bity z kolejnego znaku, kolejne trzy i tak dalej. W konsekwencji znak ósmy zostanie całkowicie usunięty. Dla dziewiątego znaku proces rozpocznie się od początku (**tab. 3**).

03	91	0100	11	00	09	91
Długość pola SCA	Typ SCA	Pole numeru SCA	Pierwszy oktet	Numer odniesienia dla wiadomości SMS.	Długość numeru odbiorcy	Typ numeru
Długość pola numeru centrum usług w bajtach (można podać 00)	91 – numeracja z numerem kierunkowym kraju, 81 – bez numeru kierunkowego	0100 = 1000 (zamieniona pozycja cyfr)	Pierwszy oktet statusu wiadomości SMS (11H to wartość typowa dla większości wiadomości)	Należy wprowadzić wartość 0	Długość numeru odbiorcy w bajtach	Podobnie jak w przypadku typu SCA – 91 lub 81

Rys. 3. Przykład ilustrujący przesłanie wiadomości o treści „THE BIG BROWN FOX” pod numer telefonu +991234567

**Tab. 2. Ilustracja zasady przekodowywania septetów na oktety**

E	L	E	K	T	R	O	N	I	K	A
0x45	0x4C	0x45	0x4B	0x54	0x52	0x4F	0x4E	0x49	0x4B	0x41
1000101	1001100	1000101	1001011	1010100	1010010	1001111	1001110	1001001	1001011	1000001
1000101	1001100	1000 <b>101</b>	1001 <b>011</b>	1010 <b>100</b>	1010 <b>010</b>	1001 <b>111</b>	1001 <b>110</b>	1001001	1001011	1000001

**Tab. 3.**

01000101	01100110	01110001	01001001	10010101	00111110	10011101	11001001	01100101	00010000
0x45	0x66	0x71	0x49	0x95	0x3E	0x9D	0xC9	0x65	0x10

Zakodowaną wiadomość „ELEKTRONIKA“ wyślijmy na numer sieci Plus GSM. Centrum wiadomości (SCA) ma tutaj numer +48601000310. Prześlijmy wiadomość na numer +48605010203.

1. Długość pola SCA (w tym przypadku 7 bajtów, z polem typu SCA): 07.

2. Typ SCA - numeracja międzynarodowa: 91.

3. Numer SCA: 8406010013F0.

4. Pierwszy oktet wiadomości SMS: 11.

5. Pole statusu: 00.

6. Długość numeru odbiorcy SMS (liczba bajtów): 07.

7. Typ numeru odbiorcy - numeracja międzynarodowa: 91.

8. Numer odbiorcy:

8496959192F3.

9. Identyfikator protokołu: 00.

10. Schemat kodowania: 00.

11. Okres ważności (1 tydzień): BF.

12. Długość tekstu: 0x0A.

13. Tekst: 45667149953E9DC96510.

Uff... Zbierzmy to w całość: 07918406010013F0110007918496959192F3000BF0A45667149953E9DC96510.

Niestety, oprócz znajomości długości niektórych ze składników wiadomości, aby ją wysłać musimy jeszcze znać jej całkowitą długość. W prezentowanym przykładzie wynosi ona 32 bajty. Zestaw poleceń, które musimy

wydać modemowi aby wysłać wiadomość, będzie następujący:

1. AT+CMGF=0 <CR>.

2. AT+CMGS=32 <CR>.

3. >07918406010013F0110007918496959192F3000BF0A45667149953E9DC96510 <CTRL+Z>.

Uwaga: wyżej wymienione znaki są reprezentacją bajtów o wartości zakodowanej szesnastkowo. Cyfra „32“ jest długością przesyłanej informacji podaną w bajtach. Podany przykład kodowania jest właściwy dla informacji wprowadzanej przez interfejs komunikacyjny. Inaczej wysyła się wiadomość, gdy numer Centrum Usług znajduje się na karcie, czy też gdy wysyłana jest wiadomość zapamiętana w pamięci telefonu lub karty SIM.

### Mode 1 (TEXT MODE)

W przeciwieństwie do poprzedniego, tryb 1 zwany również TEXT MODE jest bardzo łatwy w użyciu. Poniższa sekwencja poleceń powoduje wysłanie wiadomości w trybie tekstowym:

1. AT+CMGF=1 <CR>.

2. AT+CMGS=„+48605010203“ <CR>.

3. >ELEKTRONIKA PRAKTYCZNA <CTRL+Z>.

Trybem tym posłużyłem się tworząc procedurę „send\_SMS“. Niestety, nie wszystkie telefony GSM obsługują ten tryb pracy.

### Procedura wysyłania wiadomości SMS: „send\_sms“

Na list. 5 jest kod źródłowy procedury wysyłania wiadomości SMS. Część wiadomości docierającej do adresata jest stała, zdefiniowana w programie. Część wiadomości taka, jak lokalizacja obiektu i opis czujnika jest zmienna i może być zdefiniowana przez użytkownika. Dane te zostają zapamiętane w pamięci EEPROM i są używane przy komponowaniu wiadomości.

Do modemu jako pierwsze jest wysyłane polecenie zapisane w pamięci ROM, zdefiniowane pod etykietą *send\_sms\_1*. Jest to polecenie załączenia trybu tekstowego „AT+CMGF=1“. Uwaga! Zarówno polecenie wyświetlające listę dostępnych trybów jak i załączające konkretny tryb pracy działa dopiero po zalogowaniu się telefonu do sieci. Każde polecenie musi być zakończone znakiem CR (0x0D). Za przesłanie polecenia z ROM odpowiedzialna jest funkcja o nazwie *putf*. Różni się ona od *puts* tym, że dane wysyłane przez UART pobiera z ROM, a nie bufora w pamięci RAM. W następnej kolejności przesyłany jest do modemu numer telefonu, którego stała część zdefiniowana jest pod etykietą *send\_sms\_2*, a zmienna, wprowadzana przez użytkownika,

99214365F7	00	00	FF	11	546411244C1E4142E9F3EA94199F58
Numer odbiorcy wiadomości SMS	Identyfikator protokołu	Schemat kodowania danych	Okres ważności SMS	Długość tekstu	Tekst
Numer odbiorcy SMS: cyfry są zamieniane miejscami a w przypadku gdy zawiera on nieparzystą liczbę cyfr, na końcu dodawane jest F. (123 → 123F → 21F3)	0x00 – dostarcz jako normalny SMS	0x00 – alfabet domyślny, 7-bitowe znaki ASCII	0xFF = 255 – 192 tygodnie = 63 tygodnie	Szesnastkowo podana liczba bajtów oktetów tekstu	THE BIG BROWN FOX

Rys. 3 - cd.

pobierana jest z EEPROM. Polecenie zawierające numer telefonu adresata wiadomości komponowane jest w pamięci RAM i przesyłane do modemu przez funkcję *puts*. Podobnie tworzona jest wiadomość tekstowa: komponowana w RAM z części stałej *send\_sms\_3*, tekstu o lokalizacji obiektu wprowadzonego przez użytkownika oraz etykiety czujnika wyliczanej na podstawie zawartości zmiennej *sensors* (zmienna ta przesuwana jest w prawo z użyciem flagi przeniesienia C do momentu, aż flaga zostanie wyzerowana; logiczne „0” oznacza stan aktywny czujnika - liczba przesunięć wskazuje na numer wejścia). Tekst musi być zakończony znakiem CTRL+Z. Jest to znak końca zbioru tekstowego o kodzie 26 (0x1A).

**Jacek Bogusz, AVT**  
**jacek.bogusz@ep.com.pl**

*Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/?pdf/pazdziernik03.htm> oraz na płycie CD-EP10/2003B w katalogu PCB.*

List. 5. Procedura wysyłania wiadomości tekstowej SMS

```
send_sms:
    ldi z1,LOW(send_sms_1<<1)      ;Z = adres nastawy-polecenia w ROM
    ldi zh,HIGH(send_sms_1<<1)
    rcall putf                      ;wysłanie polecenia
    rcall gets                      ;pobranie odpowiedzi modemu
    ldi z1,LOW(send_sms_2<<1)      ;Z = adres napisu w ROM
    ldi zh,HIGH(send_sms_2<<1)
    ldi y1,LOW(buffer)             ;Y = adres bufora w RAM
    clr yh                          ;tu adresy RAM mogą być tylko 1-bajtowe
    rcall move_rom2ram             ;przepisanie danych z ROM do RAM,
    ;adresy ROM w Z, RAM w Y
    ldi eeaddr,SMSNUMBER_EEADDR    ;teraz pobranie numeru telefonu z eeprom do „buffer”,
    ;na który wskazuje Y
    rcall ee_read2buf              ;uwaga: zmieniany jest rejestr Y

    ldi temp,0x22                  ;cudzysłów zamykający
    st y,temp
    rcall st_EOL                   ;zapisanie znaków CR+LF do bufora
    rcall puts_buffer              ;wysłanie zawartości „buffer” przez UART
    ;teraz kompozycja SMS do wysłania
    ldi z1,LOW(send_sms_3<<1)
    ldi zh,HIGH(send_sms_3<<1)
    ldi y1,LOW(buffer)
    clr yh
    rcall move_rom2ram
    ldi eeaddr,SMSTEXT_EEADDR      ;teraz odczyt opisu obiektu (30 znaków wprowadzone
    ;przez użytkownika)
    rcall ee_read2buf
    ldi temp,', '                  ;do temp kod przecinka
    st y,temp
    clr temp                       ;zapisanie go w buforze
    ;wyliczenia adresu etykiety i jej odczyt (jeśli czujnik
    ;jest aktywny, to bit
send_sms_sen:
    inctemp                        ;przyjmuje wartość „0”)
    clc                             ;kasowanie flagi „carry”
    ror sensors                     ;przeniesienie najmłodszego bitu do C
    brcs send_sms_sen              ;jeśli C jest „1”,to wejście nie jest aktywne
    moveeaddr,temp
    rcall sen_ee_addr               ;wyznaczenie adresu w eeprom
    incyl                           ;odczyt etykiety wejścia do bufora w RAM
    rcall ee_read2buf
    ldi temp,0x1A                  ;zapamiętanie znaku „Ctrl+Z”
    st y,temp
    rcall st_EOL                   ;zapisanie kodów CR+LF
    rcall puts_buffer              ;wysłanie zawartości „buffer” przez UART
    incsmsent                       ;ustawienie znacznika wysłania SMS
    ret
```