

MikroC biblioteka za PDU format SMS poruke

Saša Vučićević

Student prvog ciklusa studija
Elektrotehnički fakultet

Istočno Sarajevo, Republika Srpska, Bosna i Hercegovina
vucicevicsasa@yahoo.com

Sadržaj—U radu je opisana MikroC-ova biblioteka za dekodiranje PDU formata SMS poruke. Ova biblioteka nije postojeća biblioteka u programskom jeziku MikroC, već biblioteka realizovana u sklopu datog rada. Objasnjen je PDU format, kao i funkcije koje su se koristile u biblioteci. Date funkcije služe za lakše slanje poruka, bez pretjeranog poznavanja PDU formata. Prikaz rezultata je dat u kompajleru programskog jezika C. Prikazana je i praktična primjena date biblioteke.

Ključne riječi- SMS; PDU format; MikroC biblioteka

I. UVOD

SMS poruke je moguće slati na dva načina: u PDU formatu i Text formatu. Važno je napomenuti da su sve SMS poruke, ustvari, zapisane u PDU formatu, dok samo neki mobilni uređaji imaju mogućnost prevodenja PDU formata u Text format. Text format je jednostavniji za upotrebu, ali njegova mana je što ga podržava mali broj mobilnih uređaja. Za razliku od njega, PDU format je složeniji, ali je podržan od strane svih mobilnih uređaja.

Zbog toga je kreirana biblioteka koja služi za prevodenje ovog formata u jednostavniji oblik razumljiv svakom korisniku. Date funkcije unutar biblioteke omogućavaju slanje SMS-a unošenjem broja telefona na koji se želi poslati poruka i unošenjem sadržaja poruke. Takođe, date su i funkcije koje omogućavaju čitanje poruke u Text formatu, a koja je primljena u PDU formatu.

Rad svih ovih funkcija je testiran u Microsoft Visual Studio-u, programskom jeziku C, radi jednostavnijeg kompajlera u odnosu na programski jezik MikroC, a način pisanja programa je identičan.

II. PDU FORMAT

PDU format ili protocol data unit predstavlja informaciju, koja se dostavlja kao cjelina kroz nekoliko entiteta mreže i koja može sadržati kontrolnu informaciju, informaciju adrese ili informaciju o podacima. U SMS-u, PDU format, uz poruku, sadrži i dodatne informacije o poruci, onome ko šalje poruku, vremenu kada je poslata poruka i drugo. Format je zapisan u formi heksadecimalnih okteta ili decimalnih semi-okteta. Budući da je sam PDU format jako opširna tema, ovdje će se na primjeru objasniti najčešće korišteni oblik (Tabela I) [1].

Grupa okteta prikazana u tabeli sastoji se od tri dijela:

- Informacija o broju servisnog centra (SMSC – service center number);
- SMSC broj i
- SMS_DELIVER dio poruke.

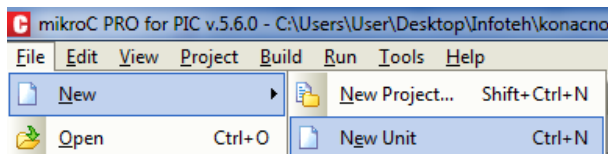
TABELA I. OBJAŠNENJE POJEDINIH DIJELOVA PDU FORMATA [1]

07	Dužina SMSC informacije (07 okteta)
91	Tip SMSC adrese (91 označava internacionalni broj, predznak +)
72 83 01 00 10 F5	SMSC broj (decimalni semi-okteti). Telefonski broj je zapisan u big endian zapisu, pa se svaka dva broja zamijene i dobije se broj "+27381000015"
04	Prvi oktet SMS poruke
0B	Dužina broja onoga koji šalje poruku (0B hex = 11 dec)
C8	Tip adrese sa koje se šalje poruka
72 38 88 09 00 F1	Broj sa kojeg se šalje poruka (decimalni semi-okteti), big endian ("27388890001")
00	TP-PID. Identifikator protokola
00	TP-DCS. Način kodiranja podataka (7-bitni podaci)
99 30 92 51 61 95 80	TP-SCTS. Vremenska oznaka (semi-okteti), 99/03/29 15:16:59 +2GMT
0A	TP-UDL. Dužina poruke, broj septeta (10).
E8329BFD4697D9EC37	TP-UD. Poruka "hellohello", 8-bitni okteti predstavljaju 7-bit podatke.

Polje za poruku u PDU formatu se sastoji od heksadecimalnih okteta, ali ovi okteti predstavljaju 7-bitne podatke. Razlog kodiranja u 7-bitne podatke je taj što se u SMS porukama ne koriste svi znakovi iz ASCII tabele (256 znakova), nego samo slova engleske abecede i neki posebni interpunkcijski znakovi (128 znakova). Time se na nizu od osam znakova, kodiranjem 7-bitnih znakova u osam bita, uštedi jedan oktet [1].

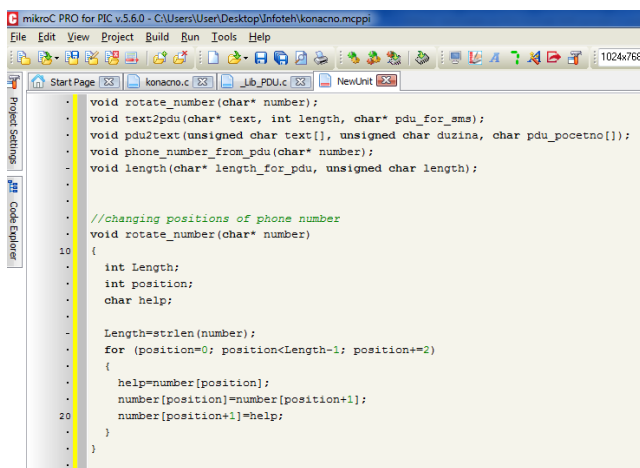
III. KREIRANJE MIKROC-OVE BIBLIOTEKE

Primjer kreiranja MikroC biblioteke dat je na stranici Mikroelektronike [2]. Prvo je potrebno kreirati novi C source fajl. Source fajlovi koji sadrže C kod treba da imaju ekstenziju .c. Da bi se kreirao novi fajl, potrebno je odabrati File->New Unit, kao što je prikazano na slici 1 [3].



Slika 1. Kreiranje novog fajla

U prozoru koji je prikazan na slici 2, potrebno je unijeti kod sa željenim funkcijama, koje će kasnije činiti biblioteku.



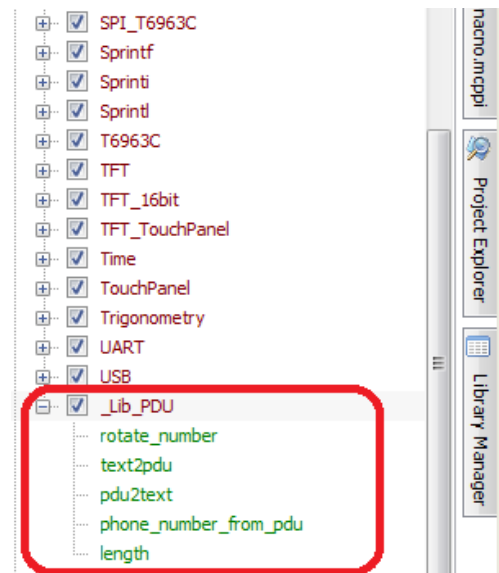
Slika 2. Unošenje funkcija

Ovaj fajl je potrebno sačuvati unutar foldera Mikroelektronika/Uses/P16 ili P18, u zavisnosti od toga za koje mikrokontrolere želimo da koristimo biblioteku [2]. Fajl je sačuvan pod imenom _Lib_PDU i sledeći korak je dodati ovaj fajl u neki projekat u MikroC-u i izvršiti kompajliranje [2]. U folderu, u kojem je sačuvan dati fajl, trebao bi se pojaviti _Lib_PDU.mcl fajl [2]. Nakon ovog koraka, potrebno je otvoriti folder Mikroelektronika/Defs i unutar njega otvoriti fajl sa ekstenzijom .mlk mikrokontrolera unutar kojeg se želi koristiti biblioteka [2].

U tom fajlu je potrebno unijeti sledeće:

```
<LIB>
<ALIAS>PDU</ALIAS>
<FILE> Lib PDU</FILE>
<TYPE>REGULAR</TYPE>
</LIB> [2].
```

Dati fajl se sačuva i u projektu sa odgovarajućim mikrokontrolerom se može koristiti data biblioteka (slika 3).



Slika 3. Prikaz date biblioteke u projektu

IV. OBJAŠNJENJE RADA POJEDINIH FUNKCIJA UNUTAR BIBLIOTEKE

Biblioteka _Lib_PDU se sastoji od sledećih funkcija:

- void rotate_number(char* number);
- void text2pdu(char* text, int length, char* pdu_for_sms);
- void pdu2text(unsigned char text[], unsigned char duzina, char pdu_pocetno[]);
- void phone_number_from_pdu(char* number);
- void length(char* length_for_pdu, unsigned char length).

Treba napomenuti da se u biblioteci nalaze funkcije koje vrše kodiranje promjenljivih dijelova PDU formata, kao što su sadržaj poruke, dužina poruke i broj pošiljaoca, kao i broj servisnog centra. Konstantni dijelovi PDU formata nisu uzeti u obzir, jer i nisu od neke važnosti za sadržaj poruke. Takođe, informacija o vremenu slanja poruke je izbačena, jer mikrokontroler nema internu mogućnost određivanja vremena. Potrebno je objasniti značenje svake od datih funkcija unutar _Lib_PDU biblioteke.

Funkcija **rotate_number** se koristi u slučaju slanja poruke sa mobilnog uređaja. Ova funkcija služi za rotiranje po dvije cifre broja na koji se želi poslati poruka, kao i broja servisnog centra. Zbog neparnog broja cifara (11 cifara), potrebno je, unutar funkcije, dodati znak 'F' na kraj broja. Rezultat ove

funkcije se smješta na odgovarajuće mjesto unutar PDU formata.

Funkcija **text2pdu** služi za konverziju sadržaja poruke iz tekstualnog formata u PDU format, koji se poslije smješta na odgovarajuće mjesto. Promjenljivi dijelovi ove funkcije su `text`, `length` i `pdu_for_sms`. `Text` i `length` se unose od strane korisnika, dok se promjenljiva `pdu_for_sms` dobija i postavlja na odgovarajuće mjesto u PDU formatu.

Funkcija **pdu2text** vrši inverziju u odnosu na prethodnu funkciju. Ona se koristi pri čitanju primljene poruke. Poruka je primljena u PDU formatu i iz nje je potrebno izdvojiti dio koji predstavlja sadržaj poruke (`pdu_real`) i konvertovati ga datom funkcijom u tekst formu (`text`).

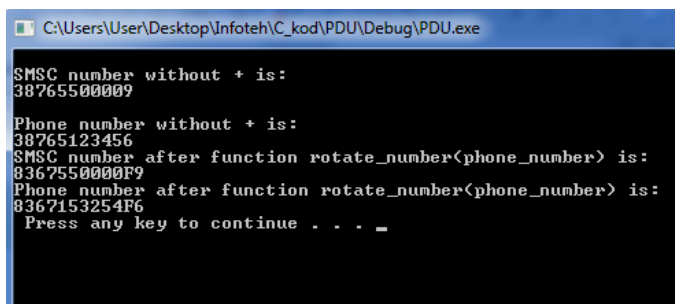
Funkcija **phone_number_from_pdu** ima za zadatak da iz PDU formata primljene poruke izvuče SMSC broj, kao i broj pošiljaoca poruke, koji su okrenuti i onda ih vrati u normalno stanje, u kojem se može prepoznati koji je broj servisnog centra i koji je broj pošiljaoca poruke. Dakle, ona vrši suprotnu operaciju od funkcije `rotate_number`.

Funkcija **length** služi za određivanje dužine poruke koja se šalje. Ona je značajna, jer informacija o dužini poruke se nalazi unutar PDU formata. Bitno je naglasiti da ova dužina ima heksadecimalnu vrijednost, zbog čega je i potrebno koristiti datu funkciju (za izvršenje konverzije iz decimalnog u heksadecimalni format).

V. PRIKAZ REZULTATA RADA POJEDINIH FUNKCIJA IZ BIBLIOTEKE `_LIB_PDU` U MICROSOFT VISUAL STUDIO-U

Provjera rada funkcija iz biblioteke `_Lib_PDU` izvršena je u Microsoft Visual Studio-u, programskom jeziku C. Kod koji se koristi je identičan kao u Mikro-C-u, s tim što je ovdje korištena `printf` funkcija za prikaz rada datih funkcija u kompajleru.

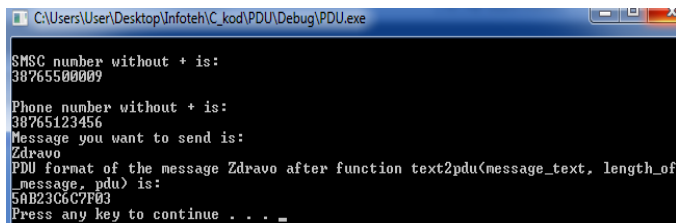
Rad funkcije **rotate_number** je testiran unosom broja +38765500009 za broj servisnog centra i unosom broja +38765123456 za broj na koji se šalje poruka. Rezultat rada ove funkcije je prikazan na slici 4.



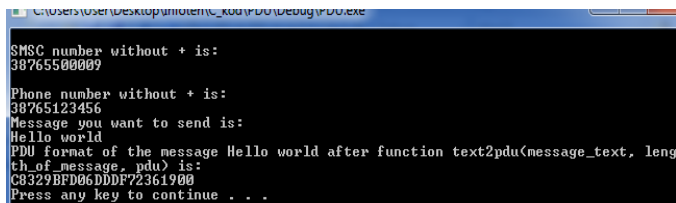
Slika 4. Prikaz rada funkcije `rotate_number`

Funkcija **text2pdu** je testirana na više primjera, a njen rad će biti prikazan na primjerima slanja sledećih poruka:

Zdravo i Hello world (slika 5).



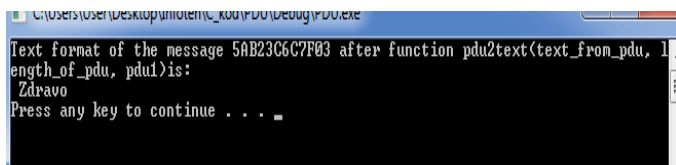
a)



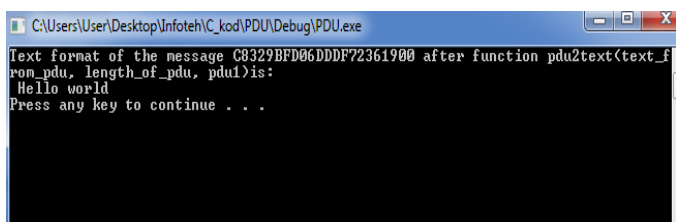
b)

Slika 5. Prikaz rada funkcije `text2pdu`: a) Zdravo; b) Hello world

Funkcija **pdu2text** treba da izvrši suprotnu operaciju u odnosu na prethodnu funkciju. Uzeta su dva ista primjera kao i u slučaju funkcije `text2pdu`, s tim što je sad dat PDU format poruke, a trebaju se dobiti poruke Zdravo i Hello world (slika 6).



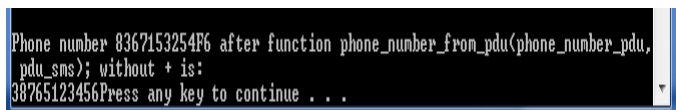
a)



b)

Slika 6. Prikaz rada funkcije `pdu2text`: a) Zdravo; b) Hello world

Rad funkcije **phone_number_from_pdu** je testiran unosom broja +38765123456 za broj sa kog je stigla poruka. Ovaj broj u PDU format ima sledeći izgled: 83 67 15 32 54 F6. Upotrebom funkcije dobija se prvobitni broj, koji predstavlja stvarni broj sa koga je poruka poslata (slika 7).



Slika 7. Prikaz rada funkcije `phone_number_from_pdu`

Funkcija **length**, kao što je već rečeno daje dužinu poruke u heksadecimalnoj vrijednosti. Rad ove funkcije je prikazan na slici 8, gdje vraća dužine poruka koje smo već do sada testirali.

```

Message you want to send is:
Zdravo
Duzina poruke zdravo je:
06Press any key to continue . . . _

```

a)

```

Message you want to send is:
Hello world
Duzina poruke Hello world je:
06Press any key to continue . . .

```

b)

Slika 8. Prikaz rada funkcije length: a) Dužina poruke Zdravo; b) Dužina poruke Hello world.

VI. PRAKTIČNA PRIMJENA BIBLIOTEKE

Biblioteka služi za slanje/primanje SMS poruka uz pomoć mikrokontrolera. Prvenstveno je potrebno ostvariti konekciju između mobilnog uređaja i mikrokontrolera. Mikrokontroler se isprogramira u programskom jeziku MikroC da izvrši željenu operaciju. U okviru ovog rada bilo je potrebno pročitati primljenu poruku sa mobilnog telefona i na osnovu pročitane poruke poslati odgovarajuću poruku uz pomoć programa mikrokontrolera. Da bi se sve ovo moglo izvršiti potrebno je koristiti AT komande.

AT komande se koriste za kontrolu modema. Uređaji koji obuhvataju komunikaciju između dvije mašine zahtijevaju AT komande za interakciju sa računarom. AT komande sa GSM/GPRS modemom ili mobilnim telefonom se mogu koristiti za pristup sledećim informacijama i servisima:

- 1) Informacija i konfiguracija odnosa modema i SIM kartice;
- 2) SMS servisi;
- 3) MMS servisi;
- 4) Fax servisi;
- 5) Razmjena podataka preko mobilne mreže [4].

Postoje četiri tipa AT komandi:

- 1) Test komande – koriste se za provjeru podržavanja date komande na modemu;
- 2) Komande za čitanje – koriste se za dobijanje podešavanja operacija modema;
- 3) Komande za postavljanje – koriste se za modifikovanje podešavanja operacija modema;
- 4) Komande za izvršavanje – koriste se za izvršavanje određenih naredbi [4].

U ovom radu su potrebne AT komande za čitanje i slanje poruka, pa u nastavku slijedi objašnjenje datih komandi. Naredba AT+CMGL[=status_poruke] služi za čitanje poruka sa mobilnog telefona. Sintaksa ove naredbe za PDU format poruke je AT+CMGL[=status_poruke]. Status poruke je dat jednim brojem od 0 do 4 i svaki od tih brojeva služi za čitanje određenih poruka. Značenje datih brojeva je sledeće: 0 –

primljene nepročitane poruke; 1 – primljene pročitane poruke; 2 – sačuvane neposlate poruke; 3 – sačuvane poslate poruke i 4 – sve poruke. Pored čitanja poruka, potrebno je omogućiti i slanje poruka sa uređaja. To se ostvaruje naredbom AT+CMGS. Ova komanda služi za slanje poruka preko računara, data poruka neće ostati sačuvana u GSM uređaju. Sintaksa ove naredbe je sledeća: AT+CMGS=dužina_pdu_poruke [5].

Nakon upoznavanja sa AT komandama, moguće je pristupiti realizaciji programa mikrokontrolera. U ovom slučaju iskorišten je mikrokontroler PIC18F2550. Pored biblioteke _Lib_PDU, bilo je potrebno koristiti i naredbe iz postojeće biblioteke UART. Mikrokontroler je isprogramiran tako da pročita poruku iz mobilnog telefona i poredi je sa tri željene poruke (definisane u programu mikrokontrolera). U slučaju da je poruka identična nekoj od željenih poruka, mikrokontroler će izdati naredbu za slanje određene poruke na određeni broj telefona, koji su definisani programom mikrokontrolera. Dio koda je prikazan na slici 9, a u svrhu provjere rada biblioteke razvijena je i štampana pločica sa mikrokontrolerom PIC18F2550, koja je prikazana na slici 10.

```

StartPage  StartPage  MyProject.c
Delay_ms(2000);
PORTB=0b00000001;
while(1)
{
140  if(UART_Data_Ready())
    {
        UART1_Write_Text("AT+CMGL=4\r\n");
        UART1_Read_Text(poruka, "OK", 255);
        Delay_ms(2000);
        UART1_Write_Text(poruka);
        Delay_ms(2000);

        for(i=0; i<6;i++)
150          pdu_ispis[i]=poruka[84+i];
        pdu2text(text4, duzina_pdu, pdu_ispis);
        UART1_Write_Text(text4);
153      for(i=0; i<4;i++)
        text5[i]=toupper(text4[i]);
        UART1_Write_Text(text5);
        UART1_Write_Text("\r\n");
        if(strcmp("2DR", text5)==0)

```

Slika 9. Dio koda mikrokontrolera



Slika 10. Štampana pločica sa isprogramiranim mikrokontrolerom

VII. ZAKLJUČAK

U ovom radu je detaljno opisana biblioteka `_Lib_PDU`. Biblioteka služi za lakše korišćenje PDU formata SMS poruke, bez pretjeranog poznavanja datog formata. PDU format je univerzalniji od Text formata SMS poruke, tako da je potrebno češće korišćenje PDU formata.

Formirano je pet funkcija koje se nalaze u sklopu biblioteke `_Lib_PDU`. Ove funkcije su kreirane u programskom paketu MikroC i imaju ekstenziju.c. Uz pomoć datih funkcija može se izvršavati dekodiranje svih potrebnih dijelova PDU formata SMS poruke:

- Konvertovanje broja pošiljaoca poruke;
- Pretvaranje poruke iz tekstualnog formata u PDU format;
- Pretvaranje poruke iz PDU formata u tekstualni format;
- Pronalaženje dužine poslate poruke.

Dijelovi PDU formata koji su konstantni nisu uzimani u obzir. Takođe, nije napravljena funkcija za konverziju vremena slanja poruke, jer mikrokontroler interno nema mogućnost poznavanja vremena.

Rad svih funkcija je provjeren u Microsoft Visual Studio-u radi lakšeg kompajliranja i korištenja funkcija koji omogućavaju ispisivanje dobijenih rezultata. Rad date funkcije je, takođe, ispitan na praktičnom primjeru primanja i slanja SMS poruka.

ZAHVALNICA

Istraživanja u ovom radu rađena su u sklopu izrade seminarskog rada iz predmeta „Mikroprocesorski sistemi“ na prvom ciklusu studija na Elektrotehničkom fakultetu u Istočnom Sarajevu, pod mentorstvom asistenta Ognjena Bjelice.

LITERATURA

- [1] SMS and The PDU format, http://subnets.ru/saved/sms_pdu_format.html, 12 Decembar 2012, 12h
- [2] Creating New Library, http://www.mikroe.com/esupport/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=38, 15 Decembar 2012, 13h
- [3] Source Files, http://www.mikroe.com/esupport/index.php?_m=knowledgebase&_a=viewarticle&kbarticleid=32, 15 Decembar 2012, 14h
- [4] AT commands, <http://www.engineersgarage.com>, 13 Decembar 2012, 18h
- [5] Developer's home, www.developershome.com, 13 Decembar 2012, 19h

ABSTRACT

This paper describes use of the PDU library, made by author of the paper, in program language MikroC. This library consists of functions which purpose is to decode PDU format into Text Format of SMS message. It is important because most of the cell phones does not support Text Format of the SMS message. There are five functions in the library which is called `_Lib_PDU`. Those functions works following things: convert Text format to PDU format, convert PDU format to Text format, rotate every two digits from cell phone number and gets length of the message.

MikroC library for PDU format of the SMS

Saša Vučićević