

Programska podrška za GPS/GSM/GPRS module

Letić, Sandi

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:964538>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-02**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Sveučilišni studij računarstva

Programska podrška za GPS/GSM/GPRS module

Završni rad

Sandi Letić

Osijek, 2018.

SADRŽAJ

1.	UVOD	1
1.1.	Zadatak završnog rada	1
2.	PRIMIENJENE TEHNOLOGIJE I ALATI	2
2.1.	A7	2
2.2.	Croduino.....	3
2.3.	Atom i Arduino IDE.....	4
2.4.	Github	5
2.5.	AT naredbe.....	6
3.	REALIZACIJA SUSTAVA.....	7
3.1.	Ideja i osnovni koncept	7
3.2.	Spajanje i priprema platforme	7
3.3.	Pristup programiranju.....	8
3.4.	Datoteka zaglavlja	8
3.5.	Implementacija funkcija.....	12
3.6.	Primjeri.....	20
3.7.	Izgled makete.....	23
4.	ZAKLJUČAK	24
	LITERATURA	25
	SAŽETAK	26
	ABSTRACT.....	27
	ŽIVOTOPIS.....	28

1. UVOD

Cilj ovog završnog rada je izrada programske podrške za GSM/GPRS/GPS module serije A6/A7/A20 proizvođača Ai-Thinker za Arduino platformu. To podrazumijeva izradu biblioteke koja bi se mogla priključiti u projekte u Arduino programskom okruženju. Uređaji korišteni u ovom slučaju imaju *micro usb* priključak, ali preko njih nije moguće uspostaviti dvosmjernu komunikaciju sa korisničkim računalom, kao ni sa Arduinom.

Rješenje za ovaj problem je korištenje AT naredbi preko posebnih ulaznih nožica uređaja. Iako su AT naredbe jako stara, zrela i dokazana tehnologija, ona nije standardizirana, tako da svaki proizvođač komponenti implementira i dodaje naredbe na svoj način, često ne obezbjeđujući dostatnu dokumentaciju.

1.1. Zadatak završnog rada

U ovom završnom radu zadatak je izrada programske podrške za GSM/GPRS/GPS module A6/A7/A20. To podrazumijeva izradu biblioteke u programskom jeziku C++, koja bi se mogla uključiti u projekte u Arduino programskom okruženju, radi lakše implementacije budućih projekata. U biblioteku je ugrađena i podrška za ESP8266 platformu, te su napravljeni kratki primjeri radi olakšanog upoznavanja sa bibliotekom.

2. PRIMIJENJENE TEHNOLOGIJE I ALATI

2.1. A7

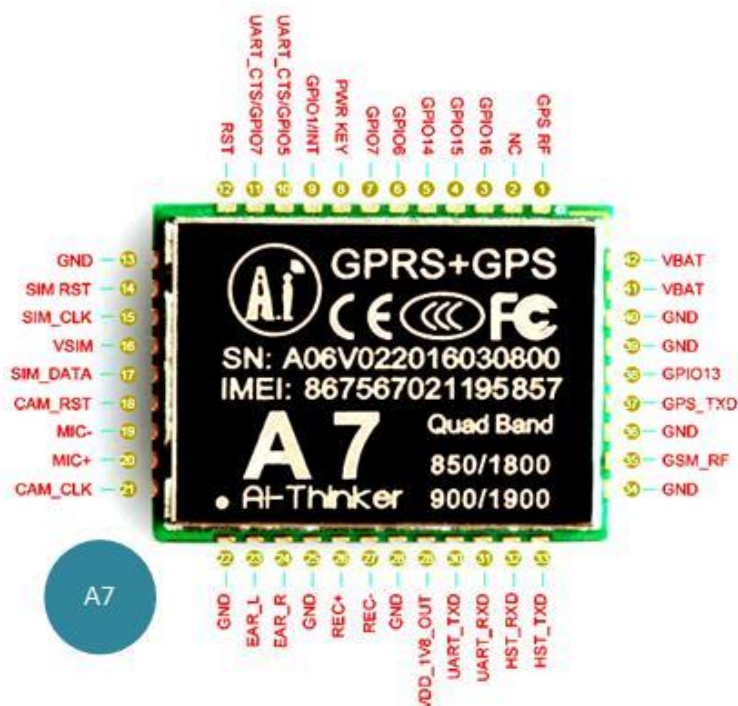
Ovaj uređaj je *chip* kineskog proizvođača Ai-Thinker iz serije A6/A7/A20 uređaja, od kojih svaki ima ugrađen utor za *SIM* karticu, *GSM* modem, pristup mobilnom internetu putem *GPRS*-a, te *GPS* prijemnik. Na priloženoj slici možemo vidjeti raspored nožica. U nastavku slijedi objašnjenje engleskih kratica:

SIM[1] - *Subscriber identity module* – jedinstvena identifikacijska kartica za mobilne telefone

GSM[2] - *Global System for Mobile communications* – mobilna mreža poznata kao *2G*

GPRS[3] – *General Packet Radio Service* – tehnologija prijenosa podatkovnih paketa vezana uz *GSM* standard

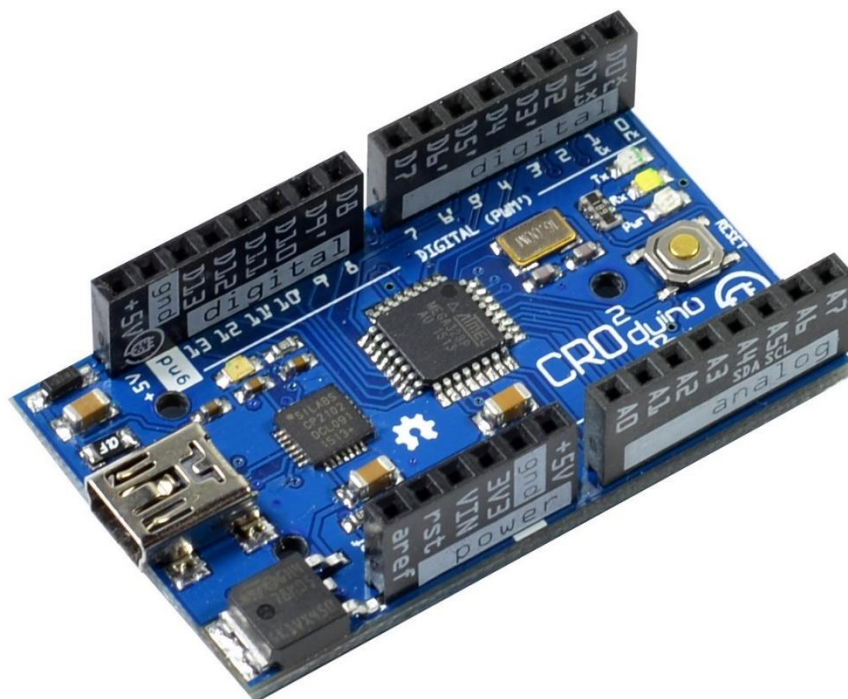
GPS[4] – *Global positioning system* – Mreža navigacijskih satelita i tehnologija preciznog određivanja lokacije



Sl. 2.1. – A7 i raspored nožica

2.2. Croduino

Croduino[5] je serija Arduino[6] kompatibilnih proizvoda dizajniranih i proizvedenih u Hrvatskoj. Arduino je platforma otvorenog koda, računalna tvrtka, zajednica i projekt koji stoji iza mikroupravljačke platforme istog imena. Arduino je postao popularan jer olakšava pristup svijetu elektronike nestručnim licima, studentima i djeci. Naglasak tvrtke proizvođača Croduina, e-radionice, je na edukaciji u školama i fakultetima i automatizaciji u industriji. Na slici 2.2 prikazan je uređaj Basic2, korišten u ovom projektu.

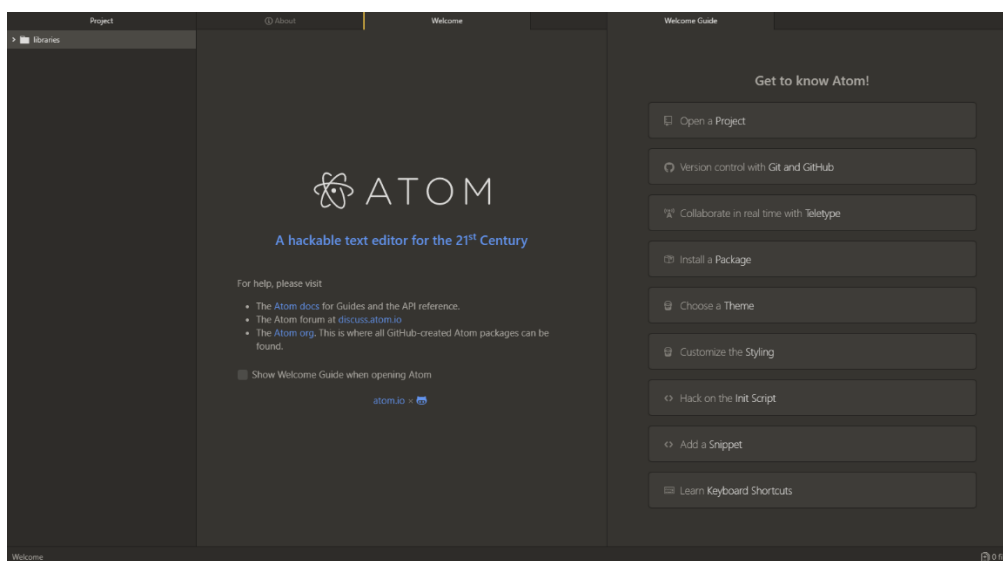


Sl. 2.2. – Croduino Basic2

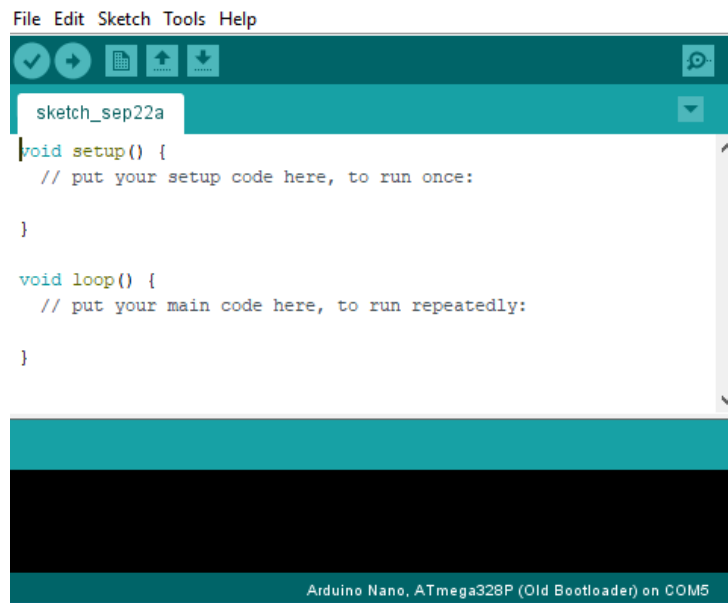
2.3. Atom i Arduino IDE

Razvojno okruženje i obradnik riječi. Kreiran je, razvijan i održavan od strane velike zajednice otvorenog koda okupljene oko GitHuba, o čemu više u sljedećem podpoglavlju. Veoma je prilagodljiv i dodatno je proširiv raznim paketima koda koji su drugi korisnici podijelili u nekom od repozitorija. Ovo razvojno okruženje korišteno je za razvoj biblioteke. Početnu stranu možemo vidjeti na slici 2.3..

Arduino IDE je razvojno okruženje namijenjeno za programiranje Arduina i sličnih uređaja. U ovom projektu kroz njega smo programirali Croduino pločicu, te je kroz njega prolazila sva komunikacija, kako za Croduino, tako i za A7. Prazan primjer vidimo na slici 2.4.



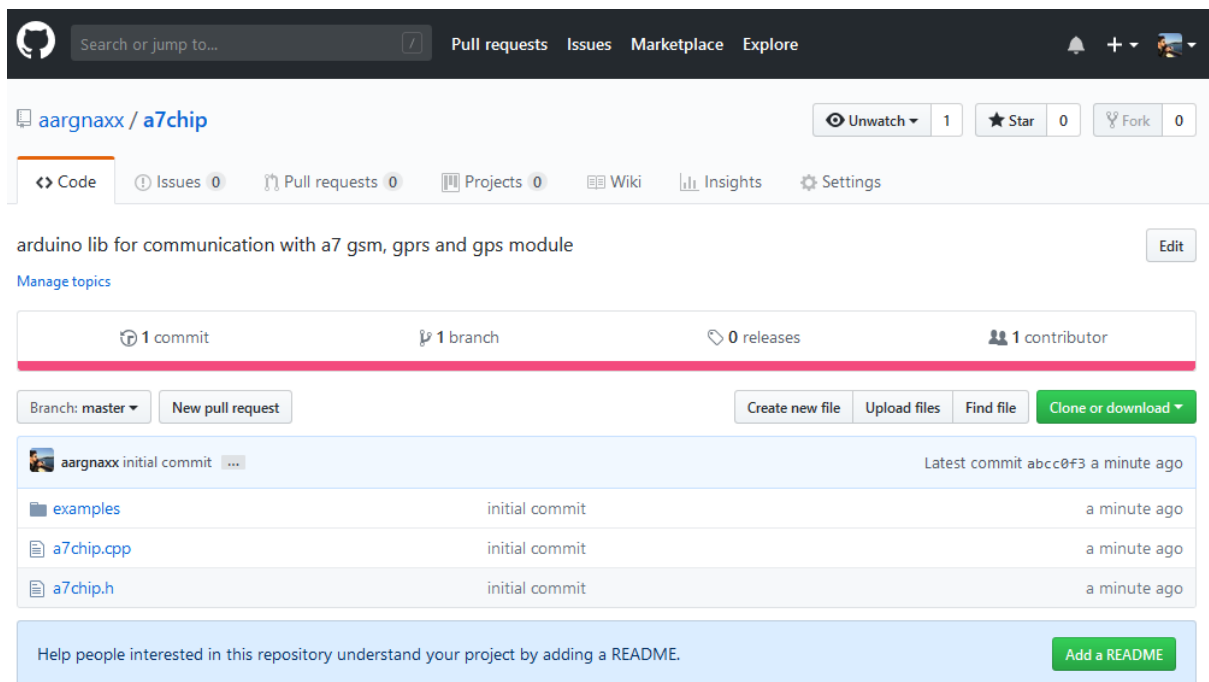
Sl. 2.3. – Atom okruženje



Sl. 2.4. – Arduino okruženje

2.4. Github

Github[7] je *online* servis za nadzor inačica, koji koristi *git*[8] sustav za upravljanje izvornim kodom. Većinom se koristi za računalni kod. Nudi sve alate git-a kao npr. raspodijeljeno upravljanje i nadzor koda, i dodaje mogućnosti poput praćenja grešaka, zahtjeve osobina (*feature*), upravljanje poslovima te informativne stranice. Na slici 2.5. možemo vidjeti repozitorij na početku projekta.



Slika 2.5. – Github repozitorij projekta

2.5. AT naredbe

Hayesov skup naredbi je naredbeni jezik, razvijen od Dennisa Hayesa u 1981. za modem 'Hayes Smartmodem 300'. Skup se sastoji od kratkih naredbenih rečenica za upravljanje pozivima, povezivanje među modemima i sl. Većina modema koristi ovaj naredbeni skup. Kako su proizvođači dodavali nove osobine modemima, tako su nadograđivali ovaj skup, tako da postoji više različitih verzija. Kratica 'AT' u AT naredbama znači *attention*, eng. za pažnju. Dodatnu specifikacija individualnih naredbi moguće je pronaći na poveznici [9].

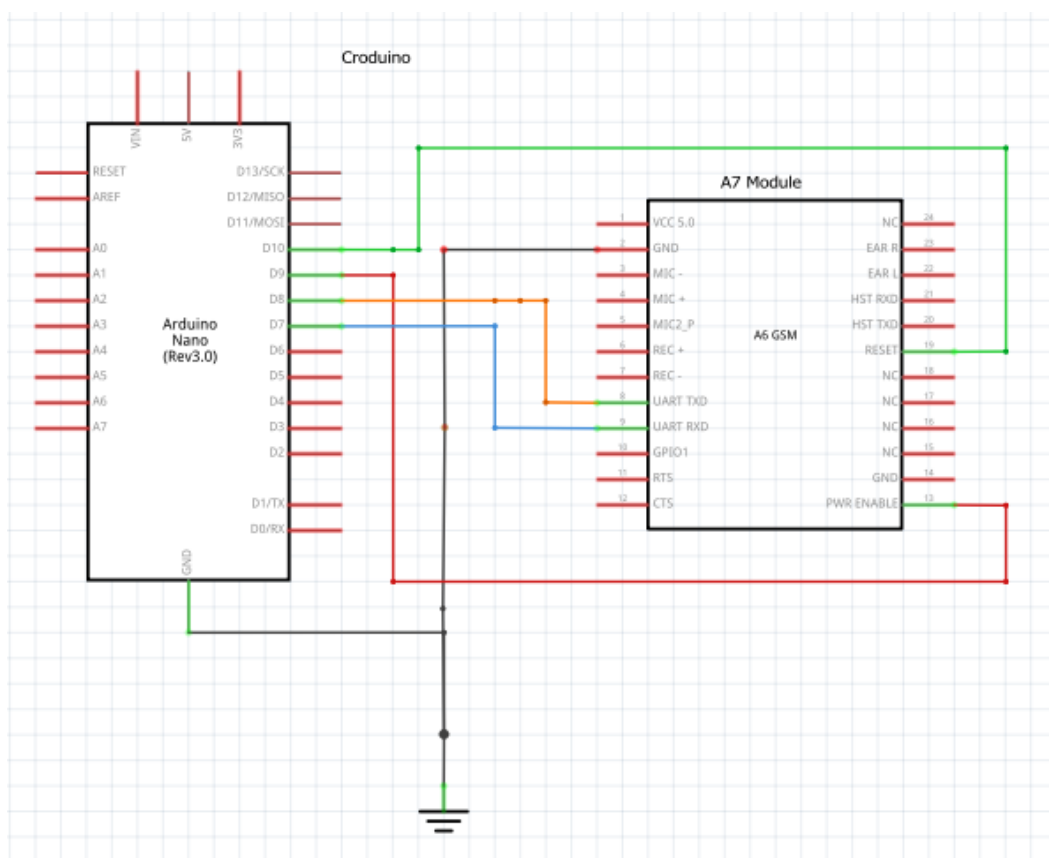
3. REALIZACIJA SUSTAVA

3.1. Ideja i osnovni koncept

Ideja iza projekta je razvoj programske podrške za A7 modul. Tome smo pristupili putem izrade biblioteke za Arduino platformu. Uređaj komunicira sa modulom koristeći AT naredbe.

3.2. Spajanje i priprema platforme

Croduino je potrebno spojiti na osobno računalo putem *USB* kabela. Potrebno je instalirati Arduino IDE te potrebne upravljačke programe za Croduino pločicu. A7 je potrebno napajati vanjskim napajanjem, također preko *USB* kabela. Može se spojiti na računalo ili na odvojen izvor napajanja. Croduino i A7 je potrebno spojiti prema dijagramu na slici 3.1. Pri spajanju nam može pomoći sljedeći priručnik[10].



Sl. 3.1. – Shema spajanja

3.3. Pristup programiranju

Unatoč nedostatku dokumentacije, moguće je naći nekolicinu github repozitorija koji se bave sličnom problematikom, primjer A[11] i B[12]. Kod pronađen u repozitorijima nije radio u praktičnom scenariju, te ga je bilo potrebno prilagoditi za potrebe ovog projekta. Repozitorij A je prilagođen mobilnim operaterima u Republici Turskoj, dok je repozitorij B pisan za upotrebu u Narodnoj Republici Kini. Repozitoriji su povezani (A je *fork*, svojevrsna kopija B), te bazirani na dokumentu izdanom od proizvođača, ali nedostupnom na njegovoj *web* stranici. Dokument je moguće pronaći ovdje [13].

3.4. Datoteka zaglavlja

Prije nego pristupimo implementaciji funkcija, potrebno ih je definirati. Prvo definiramo nožice na koje ćemo spojiti modul, zatim klasu koja predstavlja taj modul te sve potrebne metode.

```
1  /*
2  * file: AT_commands.h
3  * author: aargnaxx
4  * version: 1.0.0
5  * purpose: communication between arduino and a7 gsm module
6  * url: github.com/aargnaxx/a7chip
7  */
8
9  #include <Arduino.h>
10 #include <SoftwareSerial.h> //enable serial communication on ports not used by arduino - pc connection
11
12 #ifndef AT_commands_h
13 #define AT_commands_h
14
15 #define AT_OK 1
16 #define AT_NO 0
17 #define AT_TIMEOUT 3
18 #define AT_RST 2
19
20 #define operatorString "AT+COPS=1,0,\"bonbon\""
21 #define defaultAPN 'internet'
22
23 class a7chip {
24
25 private:
26     int pwr;
27     int rst;
28     int rx;
29     int tx;
30     char buffer[100];
31     bool debugState=false;
32
```

Sl. 3.2. – Prvi dio datoteke zaglavlja

U kodu na slici 3.2., rečenicu `operatorString` je potrebno promijeniti da odgovara izdavatelju *SIM* kartice, te unijeti odgovarajući *APN*, ukoliko se korisnik želi spojiti na Internet.

```
33 public:
34     SoftwareSerial* BoardSerial=NULL;
35
36     ~a7chip();
37     a7chip(int rxp, int txp, int pwrp=0, int rstp=0);
38     a7chip();
39
40     // function definitions
41
42     // get sim card ccid
43     String CCID();
44
45     // pin setup
46     void setup(int rx, int tx, int pwr=0, int rst=0);
47
48     // define baudrate between arduino and board
49     void begin(long baud=115200, bool reset=true);
50
51     // start the module (power pin needs to be connected to be effective)
52     void power_on();
53
54     // reset module (reset pin needs to be connected to be effective)
55     void reset();
56
57     // enter debug mode, send commands directly to the module, exit with reset
58     void serial_debug();
59
60     // start debugging information output to serial
61     void debug(bool on=true);
62
63     // execute a custom command
64     byte cmd(String command, String response1, String response2, int timeOut=500, int repetitions=2);
65
```

Sl. 3.3. – Početak javnog (*public*) dijela zaglavlja

U kodu iznad definiramo konstruktore, destruktore te metode za osnovne funkcionalnosti modula.

```

66     byte waitFor(String response1, String response2, unsigned long timeOut);
67
68     String BoardRead();
69
70     // initialize the module and prepare the AT environment
71     bool _start();
72
73     bool Operator();
74
75     bool SetAPN(String APN="defaultAPN");
76
77     // initialize the GPRS environment and start GPRS
78     bool GPRS_Start(String APN="defaultAPN");
79
80     // create a TCP link
81     bool TCP(String host, String port);
82
83     // create a UDP link
84     bool UDP(String host, String port);
85
86     // send data, link must be established (TCP or UDP)
87     bool Send(String data);
88
89     // disconnect the link (TCP or UDP)
90     bool Close();
91
92     // send data at once, including the following actions: link, send, close
93     bool Send_once(String host, String port, String data);
94
95     // simply send an at command to the module without waiting for a return
96     void at(String cmd);
97
98     // send a heartbeat pack to keep the connection alive, duration measured in seconds
99     bool heartbeat(String time="30", String send="FFFAAFFF", String get="FFFAAFFF");

```

Sl. 3.4. – Treći dio datoteke zaglavlja

U ovom dijelu definiramo metode za čitanje sa modula, osnovne opcije za korištenje podatkovnog prometa te metode za otvaranje i zatvaranje veze sa modulom.

```

104 bool TC_Start(String tries="3", String delay="300", int max_size=100, int timeout=1000);
105
106 //Stop the pass
107 bool TC_Stop();
108
109 // Transparent send data
110 bool TC_Send(String data);
111
112 // additional carriage return, waiting on TC_Start trigger condition to send, will not d
113 bool TC_Sendln(String data);
114
115 // send SMS
116 bool SendTextMessage(String Number, String msg);
117
118 // dial number
119 bool Call_Number(String Number);
120
121 // hang up the phone
122 bool Call_Number_Off();
123
124 bool debug_command (String command, unsigned long timeOut);
125
126 // GPRS functions
127 // DNS query
128 String NameToIP(String ServerName, unsigned long timeOut);
129 // regular DNS lookup
130 String NameToIP_Plus(String ServerName);
131 // HTTP get request
132 String HTTP_GET(String URL);
133 // HTTP post request
134 String HTTP_POST(String URL, String data);
135 };
136 #endif
137

```

Sl. 3.5. – Posljednji dio datoteke zaglavlja

U kodu na slici 3.5. definiramo metode za upotrebu funkcija mobilnog telefona te operacije za upotrebu TCP/IP funkcionalnosti. Pojedini komentari i prazan prostor su uklonjeni radi praktičnosti.

3.5. Implementacija funkcija

```
1  #include "a7chip.h"
2  a7chip::~~a7chip() {
3      if(BoardSerial!=NULL) {
4          delete BoardSerial;
5          BoardSerial=NULL;
6      }
7  }
8  a7chip::a7chip(int rxp, int txp, int pwrp, int rstp) {
9      a7chip();
10     setup(rx, tx, pwr, rst);
11 }
12 a7chip::a7chip() {}
13 void a7chip::setup(int rx, int tx, int pwr, int rst) {
14     rx=rx;
15     tx=tx;
16     pwr=pwr;
17     rst=rst;
18     if(BoardSerial!=NULL) {
19         delete BoardSerial;
20         BoardSerial=NULL;
21     }
22     BoardSerial=new SoftwareSerial(rx,tx);
23 }
24 void a7chip::begin(long baud, bool reset) {
25     BoardSerial->begin(baud);
26     power_on();
27     if (reset){
28         _start();
29         Operator();
30         SetAPN();
31     }
32 }
```

Sl. 3.6. – Funkcije za rad sa modulom

U priloženom kodu implementiramo destruktor i konstruktore, koji koristeći funkciju `setup()` spremaju broj nožica na koje je povezan modul. `BoardSerial` je objekt tipa `SoftwareSerial` iz istoimene biblioteke. Njega koristimo jer se Croduinov komunikacijski ulaz/izlaz već koristi za komunikaciju sa računalom, pa moramo definirati virtualno sučelje. U funkciji `begin()` koristimo neke ranije definirane metode, čiju implementaciju ćemo vidjeti naknadno.

```

33 void a7chip::power_on(){
34     if (pwr>0){
35         pinMode(pwr, OUTPUT);
36         digitalWrite(pwr, HIGH);
37         delay(500);
38         digitalWrite(pwr, LOW);
39     }
40     if (rst>0){
41         pinMode(rst, OUTPUT);
42         digitalWrite(rst, HIGH);
43         delay(500);
44         digitalWrite(rst, LOW);
45     }
46     delay(500);
47 }
48 void a7chip::reset(){
49     if (rst>0){
50         pinMode(rst, OUTPUT);
51         digitalWrite(rst, LOW);
52         delay(1000);
53         digitalWrite(rst, HIGH);
54     }
55     delay(500);
56 }
57 void a7chip::serial_debug(){
58     Serial.println("Serial Debug:");
59     while (1) {
60         if (Serial.available()) {
61             BoardSerial->write(Serial.read());
62         }
63         if (BoardSerial->available()) {
64             Serial.write(BoardSerial->read());
65         }
66     }
67 }

```

Sl. 3.7. – Upravljanje sklopovljem modula

Metode `power_on()` i `reset()` nam omogućuju upravljanje stanjem modula, ali samo u slučaju da su `pwr` i `rst` nožice spojene. `serial_debug` metoda je veoma korisna u slučaju poteškoća sa modulom, jer omogućuje direktno upravljanje i povratne informacije kroz serijsku vezu. Iz ove funkcije trenutno je moguće izaći samo ručnim resetiranjem.


```

68 void a7chip::debug(bool on){
69     debugState=on;
70 }
71 byte a7chip::cmd(String command, String response1, String response2, int timeOut, int repetitions) {
72     byte returnValue = AT_NO;
73     byte count = 0;
74     while (count < repetitions && returnValue != AT_OK) {
75         BoardSerial->println(command);
76         if (debugState){
77             Serial.print("Command: ");
78             Serial.println(command);
79         }
80         if (waitFor(response1, response2, timeOut) == AT_OK) {
81             // Serial.println("OK");
82             returnValue = AT_OK;
83         } else returnValue = AT_NO;
84         count++;
85     }
86     return returnValue;
87 }
88 String a7chip::BoardRead() {
89     String reply = "";
90     if (BoardSerial->available()) {
91         reply = BoardSerial->readString();
92     }
93     if ((debugState)&&(reply != "")) {
94         Serial.print("Reply: ");
95         Serial.println(reply);
96     }
97     return reply;
98 }
99 bool a7chip::Operator() {
100     if (cmd(operatorString, "OK", "YES", 8000, 2) == AT_OK) return AT_OK;
101     else return AT_NO;
102 }
103 bool a7chip::SetAPN(String APN) {
104     if (cmd("AT+CSSTT=\""+APN+"\", \"\", \"\", \"\", \"OK\", \"YES\", 8000, 2) == AT_OK) return AT_OK;
105     else return AT_NO;
106 }

```

Slika 3.8. – Treći dio funkcija

Metoda debug() omogućuje slanje povratnih informacija na serijsku konzolu. cmd() šalje pojedine AT naredbe modulu na siguran način. BoardRead() koristimo za čitanje odgovora modula. Operator() i SetAPN() postavljaju parametre operatora, kao što je objašnjeno u napomeni uz sliku 3.2.

```

107 byte a7chip::waitFor(String response1, String response2, unsigned long timeout) {
108     unsigned long entry = millis();
109     String reply = ""; //BoardRead();
110     byte retVal = 99;
111     do {
112         reply = BoardRead();
113         if ((debugState)&&(reply != "")) {
114             Serial.print((millis() - entry));
115             Serial.print(" ms ");
116             Serial.print("data = ");
117             Serial.println(reply);
118         }
119     } while ((reply.indexOf(response1) + reply.indexOf(response2) == -2) && millis() - entry < timeout );
120     if ((millis() - entry) >= timeout) {
121         retVal = AT_TIMEOUT;
122     } else {
123         if (reply.indexOf(response1) + reply.indexOf(response2) > -2) retVal = AT_OK;
124         else retVal = AT_NO;
125     }
126     if (debugState){
127         Serial.print("retVal = ");
128         Serial.println(retVal);
129         Serial.print("data = ");
130         Serial.println(reply);
131     }
132     return retVal;
133 }
134 bool a7chip::_start() {
135     BoardSerial->println("AT+CREG?");
136     byte hi = waitFor("1,", "5,", 1500); // 1: registered, home network ; 5: registered, roaming
137     while ( hi != AT_OK) {
138         BoardSerial->println("AT+CREG?");
139         hi = waitFor("1,", "5,", 1500);
140     }
141     if (cmd("AT&F0", "OK", "YES", 5000, 2) == AT_OK) // Reset to factory settings
142         if (cmd("ATE0", "OK", "YES", 5000, 2) == AT_OK) // disable Echo
143             if (cmd("AT+CMEE=2", "OK", "YES", 5000, 2) == AT_OK) return AT_OK; // enable better error messages
144         else return AT_NO;
145     else return AT_NO;
146     else return AT_NO;
147 }

```

Slika 3.9. – metode čekanja i pokretanja

Metoda `waitFor()` je opća metoda čekanja odgovora modula. Proširena je dodatnim uvjetima, da može vratiti odgovor modula u slučaju da je ispis omogućen.

Metoda `_start()` pokušava spojiti modul na prethodno definiranog poslužitelja te, ako to uspije, briše stare postavke, preuzima nove, te omogućuje ispis grešaka na razini 2.

```

148 bool a7chip::GPRS_Start(String APN) {
149     if (cmd("AT+CGATT=1", "OK", "YES", 8000, 2) == AT_OK) {
150         if (cmd("AT+CGDCONT=1, \"IP\", \"\"+APN+\"\", \"OK\", \"YES\", 8000, 2) == AT_OK) {
151             if (cmd("AT+CGACT=1,1", "OK", "YES", 8000, 2) == AT_OK) {
152                 if (cmd("ATD*99***1#", "CONNECT", "YES", 8000, 2) == AT_OK) {
153                     return AT_OK;
154                 }else return AT_NO;
155             }else return AT_NO;
156         }else return AT_NO;
157     }else return AT_NO;
158 }
159 bool a7chip::TCP(String host,String port) {
160     Close();
161     debug_command("AT+CIPSTATUS", 800);
162     String at="AT+CIPSTART=\"TCP\", \"\"";
163     at=at+host+"\", "+port;
164     //if (cmd(at, "CONNECT OK", "YES", 16000, 2) == AT_OK) {
165     if (cmd(at, "OK", "YES", 16000, 2) == AT_OK) {
166         return AT_OK;
167     }else return AT_NO;
168 }
169 bool a7chip::UDP(String host,String port) {
170     Close();
171     String at="AT+CIPSTART=\"UDP\", \"\"";
172     at=at+host+"\", "+port;
173     if (cmd(at, "OK", "YES", 16000, 2) == AT_OK) {
174         return AT_OK;
175     }else return AT_NO;
176 }
177 bool a7chip::Send(String data){
178     int len=data.length();
179     String sd="AT+CIPSEND=";
180     sd=sd+len;
181     sd=sd+"\", \"\"+data+\"\"";
182     // cmd("AT+CIPSEND", ">", "YES", 10000, 1); //send data to remote server
183     bool r=cmd(sd, "OK", "YES", 10000, 1); //send data to remote server
184     // delay(500);
185     // BoardSerial->print(data);
186     // BoardSerial->print(0x1A);
187     delay(600);
188     return r;
189 }

```

Slika 3.10. – GPRS metode

Metoda GPRS_start() omogućuje spajanje podatkovnim podacima. Naredbe TCP te UDP ostvaruju vezu sa poslužiteljem preko istoimenog protokola. Send() šalje jedan podatkovni paket korisničke varijable već postojećom vezom.

```

190 bool a7chip::Close(){
191     BoardSerial->print("\r\n");
192     if (cmd("AT+CIPCLOSE", "OK", "YES", 3000, 1) == AT_OK) {
193         return AT_OK;
194     }else return AT_NO;
195 }
196 bool a7chip::Send_once(String host,String port,String data){
197     if (TCP(host, port)) {
198         if (Send(data)) {
199             Close();
200             return AT_OK;
201         }else return AT_NO;
202     }else return AT_NO;
203 }
204 void a7chip::at(String cmd){
205     BoardSerial->println(cmd);
206 }
207 bool a7chip::SendTextMessage(String Number,String msg){
208     if (cmd("AT+CMGF=1", "OK", "YES", 3000, 2) == AT_OK){
209         delay(100);
210         if (cmd("AT+CMGS=\""+Number+"\"", ">", "YES", 5000, 2) == AT_OK){
211             delay(100);
212             BoardSerial->println(msg);//the content of the message
213             BoardSerial->write(0x1A);//the ASCII code of the ctrl+z is 26
214             delay(100);
215             return AT_OK;
216         }else return AT_NO;
217     }else return AT_NO;
218 }
219 bool a7chip::Call_Number(String Number){
220     cmd("AT+SNFS=0", "OK", "YES", 20000, 2);
221     if (cmd("ATD"+Number, "OK", "YES", 20000, 2) == AT_OK){
222         return AT_OK;
223     }else return AT_NO;
224 }
225 bool a7chip::Call_Number_Off(){
226     if (cmd("ATH", "OK", "YES", 20000, 2) == AT_OK){
227         return AT_OK;
228     }else return AT_NO;
229 }

```

Slika 3.11. – Daljnje metode

Metoda Close() zatvara vezu između Croduina i modula. Send_once() se ponaša kao Send(), ali nije potrebno prethodno ostvariti vezu. Naredba at() šalje AT naredbu na pomalo nesiguran način, i bez čekanja odgovora. Ne postoji garancija da će se naredba izvršiti. SendTextMessage() šalje SMS koristeći korisničke varijable. Call_Number() i Call_Number_Off() se koriste za upravljanje pozivom, ali nisu testirane, zbog nedostatka opreme.

```

230 bool a7chip::heartbeat(String time,String send,String get){
231     if (time!="0"){
232         if (cmd("AT+CIPHCFG=0,"+time, "OK", "YES", 5000, 2) == AT_OK) {
233             if (cmd("AT+CIPHCFG=1,"+send, "OK", "YES", 5000, 2) == AT_OK) {
234                 if (cmd("AT+CIPHCFG=2,"+get, "OK", "YES", 5000, 2) == AT_OK){
235                     if (cmd("AT+CIPHMODE=1", "OK", "YES", 5000, 2) == AT_OK){return AT_OK; }
236                     else return AT_NO;
237                 }else return AT_NO;
238             }else return AT_NO;
239         }else return AT_NO;
240     }else{
241         if (cmd("AT+CIPHMODE=0", "OK", "YES", 5000, 2) == AT_OK){return AT_OK; }
242         else return AT_NO;
243     }
244 }
245 bool a7chip::TC_Start(String tries,String delay,int max_size,int timeout){
246     if (cmd("AT+CIPTCFG=0,"+tries, "OK", "YES", 5000, 2) == AT_OK) {
247         if (cmd("AT+CIPTCFG=1,"+delay, "OK", "YES", 5000, 2) == AT_OK) {
248             if (cmd("AT+CIPTCFG=2,"+String(max_size), "OK", "YES", 5000, 2) == AT_OK){
249                 if (cmd("AT+CIPTCFG=3,"+String(timeout), "OK", "YES", 5000, 2) == AT_OK){
250                     if (cmd("AT+CIPTMODE=1", "OK", "YES", 5000, 2) == AT_OK){return AT_OK; }
251                     else return AT_NO;
252                 }else return AT_NO;
253             }else return AT_NO;
254         }else return AT_NO;
255     }else return AT_NO;
256 }
257 bool a7chip::TC_Stop(){
258     BoardSerial->println();
259     BoardSerial->println("+++");
260     delay(500);
261     return true;
262 }
263 bool a7chip::TC_Send(String cmd){
264     BoardSerial->print(cmd);
265     return true;
266 }
267 bool a7chip::TC_Sendln(String cmd){
268     BoardSerial->println(cmd);
269     return true;
270 }

```

Slika 3.12. – Dodatne GPRS metode

Metoda heartbeat() služi za održavanje veze sa poslužiteljem u slučaju malog protoka podataka.

TC_Start() i TC_Stop() pokreću, odnosno zaustavljaju TCP/IP 'transparentni' način.

TC_Send() i TC_Sendln() stavljaju naredbe vezane za GPRS promet, ali samo Sendln() pokreće slanje.

```

271 bool a7chip::debug_command(String command, unsigned long timeOut) {
272     BoardSerial->println(command);
273     if (debugState){
274         Serial.print("Command: ");
275         Serial.println(command);
276     }
277     unsigned long entry = millis();
278     String reply = "";
279     byte retVal = 99;
280     reply = BoardRead();
281     if (debugState){
282         Serial.print("retVal = ");
283         Serial.println(retVal);
284         Serial.print("data = ");
285         Serial.println(reply);
286     }
287     do {
288         reply = BoardRead();
289         if ((debugState)&&(reply != "")) {
290             Serial.print((millis() - entry));
291             Serial.print(" ms ");
292             Serial.print("data = ");
293             Serial.println(reply);
294         }
295     } while ( millis() - entry < timeOut );
296 }
297 //functions below not implemented
298 String a7chip::CCID(){}
299 String a7chip::NameToIP(String ServerName, unsigned long timeOut){
300     //String command = "AT+CDNSGIP=\""+ServerName+"\"";
301     String command = "AT+CDNSGIP?";
302     debug_command(command, timeOut);
303     return command;
304 }
305 String a7chip::NameToIP_Plus(String ServerName){}
306 String a7chip::HTTP_GET(String URL){}
307 String a7chip::HTTP_POST(String URL,String data){}
308

```

Slika 3.13. – debug i rezervirane naredbe

Naredba `debug_command()` omogućuje zapisivanje cjelokupne komunikacije između Croduina i modula na serijskoj konzoli. Naredbe na kraju datoteke nisu uspješno implementirane zbog nedostatka znanja i iskustva o DNS protokolu, tako da su adrese servera unesene ručno.

3.6. Primjeri

U sklopu razvoja programske podrške napravljena su tri primjera koje korisnik može isprobati.

```
1 #include "a7chip.h"
2
3 #define rx 7
4 #define tx 8
5 #define pwr 9
6 #define rst 10
7
8 a7chip Board(rx, tx, pwr, rst);
9
10 void setup() {
11     Serial.begin(115200);
12     Board.debug();
13     Board.begin(115200);
14     Board.SendTextMessage("+3859XXXXXXXX", "demo message");
15     Board.Close();
16 }
17
18 void loop(void) {}
19
```

Slika 3.14. – Primjer slanja poruke

Korisnik može unijeti željeni broj i poruku unutar parametara naredbe SendTextMessage()

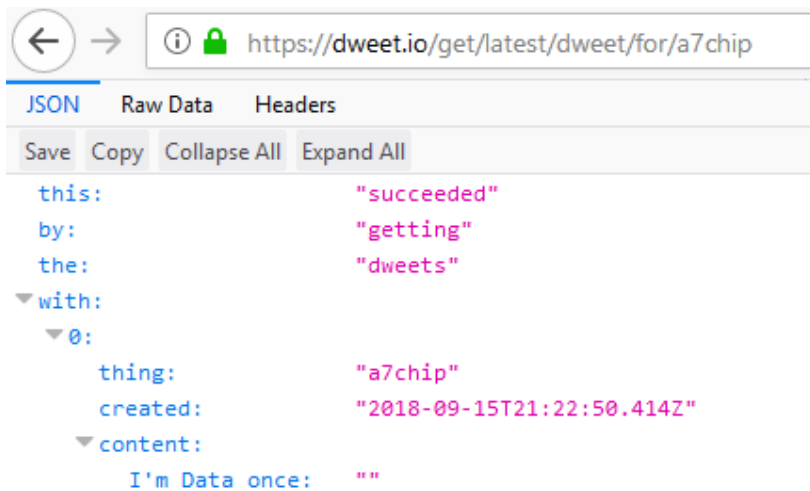
U nastavku prikazujemo primjere za komunikaciju putem mobilnih podataka.

```

1  #include "a7chip.h"
2
3  #define rx 7
4  #define tx 8
5  #define pwr 9
6  #define rst 10
7
8  a7chip Board(rx, tx, pwr, rst);
9
10 void setup() {
11     Serial.begin(115200);
12     Board.debug();
13     Board.begin(115200);
14     // start the network
15     Board.GPRS_Start();
16     // send data once:
17     Board.Send_once("https://dweet.io/dweet/for/a7chip?", "80", "I'm Data once");
18     // establish connection
19     Board.TCP("https://dweet.io/dweet/for/a7chip_rep?", "80");
20     // send data multiple times
21     for (size_t i = 0; i < 10; i++) {
22         Board.Send("I'm Data");
23         delay(2000);
24     }
25     // close the connection
26     Board.Close();
27     // using transparent mode establish connection
28     Board.TCP("https://dweet.io/dweet/for/a7chip_rep?", "80");
29     // enable heartbeat pack once every 60 seconds
30     Board.heartbeat("60");
31     // open through
32     Board.TC_Start();
33     for (size_t i = 0; i < 10; i++) {
34         Board.TC_Send("I'm Data");
35         delay(2000);
36     }
37     // exit transparent mode
38     Board.TC_Stop();
39     //shut down
40     Board.Close();
41 }
42
43 void loop(void) {}

```

Slika 3.15. – Slanje podataka korištenjem GPRS-a (TCP)



Slika 3.16. - Primjer primljenih podataka

Na slici 3.15. vidimo primjer slanja podataka putem TCP protokola. Slika 3.16. prikazuje poslani podatak na dweet.io[15] platformi. Dweet je platforma za lakše komuniciranje jednostavnih uređaja. Na slici 3.17. možemo naći sličan primjer, samo ovaj put koristeći UDP.

```

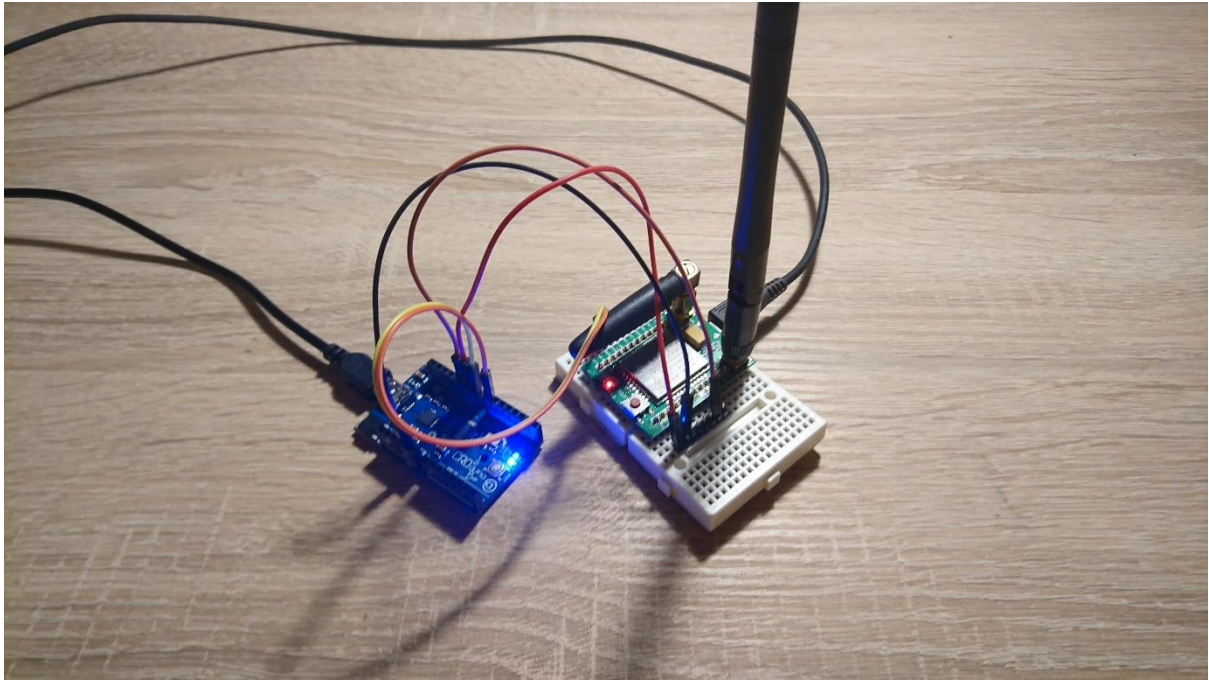
1  #include "a7chip.h"
2
3  #define rx 7
4  #define tx 8
5  #define pwr 9
6  #define rst 10
7
8  a7chip Board(rx, tx, pwr, rst);
9
10 void setup() {
11     Serial.begin(115200);
12
13     Board.debug();
14     Board.begin(115200);
15     // start the network
16     Board.GPRS_Start();
17     // send UDP data
18     Board.UDP("http://dweet.io/dweet/for/a7chip?", "80");
19     for (size_t i = 0; i < 10; i++) {
20         Board.Send("I'm Data");
21         delay(2000);
22     }
23     Board.Close();
24 }
25
26 void loop(void){}
27

```

Slika 3.17. - Slanje podataka korištenjem GPRS-a (UDP)

3.7. Izgled makete

Na slici 3.18. možemo vidjeti finalni oblik makete te operativan sklop.



Slika 3.18. – Maketa

4. ZAKLJUČAK

U ovom završnom radu napravljena je izrada programske podrške za A7 GSM/GPRS/GPS modul. Također je složena maketa za testiranje. Knjižnica i primjeri su napisani te dostupni na github repozitoriju projekta [15]. Ovaj projekt je lako proširiv i veoma modularan, te će vjerojatno podržavati mnoge nadolazeće module ovog proizvođača. Izrada nije bila jednostavna zbog manjka dokumentacije. Tehnologija AT naredbi je veoma stara i loše standardizirana i mišljenja sam da bi u budućnosti trebalo uvesti drugačiji standard. Za buduće nadogradnje biblioteke preporučujem dodavanje DNS naredbi i dodatnih primjera. Zahvaljujem mentoru, prof. Aleksiju, na ogromnom strpljenju i izlazu u susret.

LITERATURA

- [1] *3GPP specification: 51.011*. [online], <http://www.3gpp.org/dynareport/51011.htm> [17.9.2018.]
- [2] European Telecommunications Standards Institute, *GSM*. (2011). [online], <http://www.etsi.org/WebSite/Technologies/gsm.aspx> [18.9.2018.]
- [3] Howell, E. (n.d.). *Navstar: GPS Satellite Network*, SPACE.com, [online], <http://www.space.com/19794-navstar.html> [19.9.2018.]
- [4] Pérez, A. (2013). *The GPRS Network*, John Wiley & Sons, Inc., 2012.
- [5] e-radionica, O e-radionici: Naši proizvodi [online] <https://e-radionica.com/hr/about-us/#products> [17.9.2018.]
- [6] Arduino SRL, Arduino cc : Arduino – Introduction [online], <https://www.arduino.cc/en/Guide/Introduction> [19.9.2018.]
- [7] Github, Inc., github.com: Github – About [online] <https://github.com/about> [21.9.2018.]
- [8] Git community, Git SCM [online], <https://git-scm.com/> [17.9.2018.]
- [9] M2MSupport, AT Command Reference [online], <http://m2msupport.net/m2msupport/software-and-at-commands-for-m2m-modules> [21.9.2018.]
- [10] 14Core, Wiring the A7 / A6 / A6C / A2 / A20 GSM GPRS / GPS WIFI Quad Band 2G GSM Module Board, 14Core.com [online], <https://www.14core.com/wiring-the-a7-a6-a6c-a2-a20-gsm-gprs-gps-wifi-communication-module-board> [18.9.2018.]
- [11] bmericc, AiThinker_A7 [online], https://github.com/bmericc/AiThinker_A7 [15.9.2018.]
- [12] flyraining, AiThinker_A6 [online], https://github.com/flyraining/AiThinker_A6 [15.9.2018.]
- [13] Ai-Thinker, A6/A7/A6C/A20 Module AT Command Set, makerfabs.com [online], https://www.makerfabs.com/desfile/files/A6_A7_A6C_A20%20AT%20Command%20V1.03.pdf [21.9.2018.]
- [14] Bug Labs, Inc., Dweet.io, dweet.io [online], <https://dweet.io/> [21.9.2018.]
- [15] Sandi Letić, a7chip, github.com [online], <https://github.com/aargnaxx/a7chip> [23.9.2018.]

SAŽETAK

Tema projekta je bila izrada programske podrške za GSM/GPRS/GPS module. Napravljena je Arduino biblioteka u Atom okruženju i primjeri u Arduino IDE-u, te maketa i obavljeno je djelomično testiranje. Ovaj projekt podržava širok spektar modema uz određena ograničenja.

Ključne riječi: A7, Arduino, Croduino, modem, GSM, GPS, GPRS, Komunikacija, paket, Atom, git, github, Ai-Thinker, SIM, 2g, podaci, Internet, ESP8266

ABSTRACT

Title: Development of library for GSM/GPRS/GPS modules

The idea of this final paper was to make an Arduino library to support Ai-Thinker A6/A7/A20 series of GSM/GPRS/GPS modems. The library was developed in Atom. Examples were developed in Arduino IDE. A model was built and tested. The library theoretically supports a wide range of devices, but with some limitations.

Keywords: A7, Arduino, Croduino, modem, GSM, GPS, GPRS, communication, packet, data, Atom, git, github, Ai-Thinker, SIM, 2G, Internet, ESP8266

ŽIVOTOPIS

Sandi Letić rođen je 7.5.1995. u Zenici, Bosna i Hercegovina. Završio je Osnovnu školu i Opću Gimnaziju Katoličkog Školskog Centra „Sv. Pavao” Zenica, te paralelno pohađao i glazbenu školu. Ostvario je par zapaženih rezultata na općinskim natjecanjima iz matematike i informatike. Trenutno je absolvent Sveučilišnog studija Računarstva Fakulteta Elektrotehnike, Računarstva i Informatičkih tehnologija u Osijeku. Fakultet je predstavljao u natjecanju iz Objektno orijentiranog programiranja na Elektrijadi u Riminiju, Italija, 2015. godine. Akademsku 2017./18. godinu je proveo na studentskoj razmjeni u Švedskoj i Španjolskoj. Bavi se IoT-om, Linux-om, Cloud tehnologijama, mrežnom sigurnošću, programiranjem, testiranjem te 3D modeliranjem i printanjem.