

```

1  /*
2  gsm_UTILITIES v 0.921/20171130 - a-gsmII 2.105/b-gsmgnss 2.105
3  SIM/MODEM/NETWORK/POWER ON/POWER OFF utilities
4  COPYRIGHT (c) 2014-2017 Dragos Iosub / R&D Software Solutions srl
5
6  *****IMPORTANT
7  NOTICE*****
8  "agsmII_basic_lbr.h", "agsmII_SMS_lbr.ino"
9  or,
10 "bgsmsgnss_basic_lbr.h", "bgsmsgnss_SMS_lbr.ino"
11 ARE REQUIERED IN ORDER TO RUN THIS EXAMPLE!!!!!!!!!!!!!!!!!!!!!!
12 Download the "a-gsmII kickstart for Arduino"/"b-gsmgnss kickstart for Arduino" from
13 here:
14 https://itbrainpower.net/downloads
15 Uncompress the archive and copy the files mentined above in the folder
16 where is this utility, then you can compile this code.
17
18 You may like to modify the variables found at line 52
19 *****END of
20 NOTICE*****
21
22 You are legaly entitled to use this SOFTWARE ONLY IN CONJUNCTION WITH
23 a-gsmII/b-gsmgnss DEVICES USAGE. Modifications, derivates and redistribution
24 of this software must include unmodified this COPYRIGHT NOTICE. You can redistribute
25 this SOFTWARE and/or modify it under the terms
26 of this COPYRIGHT NOTICE. Any other usage may be permitted only after written notice
27 of Dragos Iosub / R&D Software Solutions srl.
28
29 This SOFTWARE is distributed is provide "AS IS" in the hope that it will be useful,
30 but WITHOUT ANY WARRANTY; without even the implied
31 warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
32
33 Dragos Iosub, Bucharest 2017.
34 http://itbrainpower.net
35 */
36 /*
37
38 In order to make the Arduino serial communication (especially for Arduino Uno) with
39 a-gsmII/b-gsmgnss shield reliable you must
40 edit C:\Program Files\Arduino\libraries\SoftwareSerial\SoftwareSerial.cpp
41 comment at line 42
42 #define _SS_MAX_RX_BUFF 64 ( will look like: //#define _SS_MAX_RX_BUFF 64 )
43 and add at next line
44 #define _SS_MAX_RX_BUFF 128
45 You just increased increase the RX buffer size speed for UNO and other snails...
46 */
47 /*
48 Eventually, you may want to test activateTopSIM() or activateBottomSIM()...just add
49 code somewhere in the loop(), at the end of one case, like:
50 case xx:
51     ....some previous code
52     activateBottomSIM();// or activateTopSIM()...Active SIM value will be stored in
53     activeSIM var
54     state=0;
55     ...in setup() function
56 Enjoy!
57 */
58 //#define atDebug
59 //next 2 definition: leave them commented for standard conectivity over Software
60 serial
61 //#define usejLader //un-comment this if you use micro and nano
62 GSM 3G adapter for ArduinoNano --Do not use it with a-gsmII/b-gsmgnss!!!!
63 //#define HARDWARESERIAL //remove comment to use Serial1 for

```

```

communication on AT MEGA 2560...DUE..
50
51 //change next to fit your destination number
52 char destinationNumber[]=""; //usually phone number with International prefix
    eg. +40 for Romania
53
54 //void activateTopSIM(void){setActiveSIM(0);}//call this function if you want to
    test/use the a-gsmII/b-gsmgnss primary SIM card
55 //void activateBottomSIM(void){setActiveSIM(1);}//call this function if you want to
    test/use the a-gsmII/b-gsmgnss second SIM card
56
57 /*do not change under this line! Instead, make one copy for playing with.*/
58 #define powerPIN      7//Arduino Digital pin used to power up / power down the modem
59 #define resetPIN     6//Arduino Digital pin used to reset the modem
60 #define statusPIN    5//Arduino Digital pin used to monitor if modem is powered
61
62 #if (ARDUINO >= 100)
63     #include "Arduino.h"
64     #if !defined(HARDWARESERIAL)
65         #include <SoftwareSerial.h>
66     #endif
67 #else
68     #include "WProgram.h"
69     #if !defined(HARDWARESERIAL)
70         #include <NewSoftSerial.h>
71     #endif
72 #endif
73
74 #if defined(HARDWARESERIAL)
75     #define BUFFDSIZE 1024
76 #else
77     #if defined(__AVR_ATmega1280__) /*AT MEGA ADK*/|| defined(__AVR_ATmega2560__)
78         /*AT MEGA 2560*/|| defined(__AVR_ATmega32U4__) /*LEONARDO*/
79         SoftwareSerial agsmSerial(10,3); //RX==>10,TX soft==>3...read
80         #define BUFFDSIZE 1024
81     #else/*UNO*/
82         #define UNO_MODE //Arduino UNO
83         #define BUFFDSIZE 200 //240
84         #if defined usejLader
85             SoftwareSerial agsmSerial(3, 2); //RX==>3 ,TX soft==>2
86         #else
87             SoftwareSerial agsmSerial(2, 3); //RX==>2 ,TX soft==>3
88         #endif
89     #endif
90 #endif
91
92 #include "bgsmgnss_basic_lbr.h"
93
94 #define printDebugLN(x){Serial.println(x);}
95
96 int state=0, i=0, powerState = 0;
97 char ch;
98 char buffd[256];
99 //char IMEI[18];
100 unsigned long offsetTime;
101 int totalChars = 0;
102 int ready4SMS = 0;
103 int ready4Voice = 0;
104 char readBuffer[200];
105
106 void setup() {

```

```

107   agsmSerial.begin(9600);
108   Serial.begin(57600);
109   clearagsmSerial();
110   clearSerial();
111   delay(10);
112
113   modemHWSetup(); //configure Arduino IN and OUT to be
   used with modem
114
115   Serial.flush();
116   agsmSerial.flush();
117   delay(1000);
118   Serial.println(F("a-gsmII/b-gsmgnss UTILITIES example"));
119   Serial.flush();
120
121   if(strlen(destinationNumber)<1){
122       Serial.print(F("destinationNumber not initialized. Edit gsm_UTILITIES_SS.ino
   and set the destinationNumber(line 41) with your phone number.\r\n\r\nNow
   the program will stop.));
123       delay(1000);
124       exit(0);
125   }
126
127   Serial.println(F("seat back and relax until a-gsmII/b-gsmgnss is ready"));
128   delay(100);
129
130   powerOnModem();
131
132   //void activateTopSIM(void){setActiveSIM(0);}//call this function if you want to
   test/use the a-gsmII/b-gsmgnss primary SIM card
133   //void activateBottomSIM(void){setActiveSIM(1);}//call this function if you
   want to test/use the a-gsmII/b-gsmgnss second SIM card
134
135   clearBUFFD();
136   while(strlen(buffd)<1){
137       getIMEI();
138       delay(500);
139   }
140
141   ready4SMS = 0;
142   ready4Voice = 0;
143
144   Serial.println(F("a-gsmII/b-gsmgnss ready.. let's run the example"));
145   Serial.print(F("a-gsmII/b-gsmgnss IMEI: ")); Serial.flush();
146   Serial.println(buffd); Serial.flush();
147
148   printDebugLN(F("\r\n\r\nHEALTH AND SAFETY WARNING!!!!!!!!!!!!!!!!!!!!!!!!!!!!"));
149   printDebugLN(F("High power audio (around 700mW RMS)! You can damage your years!
   Use it with extreme care when headset is connected.));
150   printDebugLN(F("We recomend to use maximum AT+CLVL=25, audio setup command in
   order to limit the output power.\r\n\r\n"));
151   delay(5000);
152 }
153
154 void loop(){
155     int callStatus;
156     int res;
157     int i;
158     switch(state){
159         case 0://check modem status
160             if(!getModemState()) restartMODEM();
161             else

```

```

162     state++;
163     i=0;
164     res= 0;
165     while(res != 1){
166         res = sendATcommand("", "OK", "ERROR", 2);
167         if (res != 1) {
168             if(i++ >= 10) {
169                 printDebugLN(F("AT err...restarting"));
170                 restartMODEM();
171             }
172         }
173         delay(500);
174     }
175     sendATcommand("+IPR=0;&w", "OK", "ERROR", 2);
176     delay(2000);
177     break;
178
179     case 1:
180         clearBUFFD();
181         //next some init strings...
182         aGsmCMD("AT+QIMODE=0", 200);
183         aGsmCMD("AT+QINDI=0", 200);
184         aGsmCMD("AT+QIMUX=0", 200);
185         aGsmCMD("AT+QIDNSIP=0", 200);
186         offsetTime=0;
187         clearBUFFD();
188         state++;
189         break;
190
191     case 2:
192         printDebugLN(F("try CPIN..."));
193         if(!offsetTime) offsetTime = millis();
194         if ((millis() - offsetTime) > 20000) restartMODEM();
195         if(sendATcommand("+CPIN?", "READY")==1){
196             offsetTime=0; state++;
197             printDebugLN(F("READY"));
198         }else{}
199         clearagsmSerial(); delay(100);
200         offsetTime = millis();
201         break;
202
203     case 3:
204         if(!offsetTime) offsetTime = millis();
205         if ((millis() - offsetTime) > 30000) restartMODEM();
206
207         printDebugLN(F("Query GSM registration?"));
208         res = registration(GSM);
209         if(res==1){
210             offsetTime=0; state++;
211             printDebugLN(F("READY, HOME NETWORK"));
212         }else if(res==5){
213             offsetTime=0; state++;
214             printDebugLN(F("READY, ROAMING"));
215         }else{
216             Serial.print(F("."));
217         }
218         offsetTime = millis();
219         break;
220
221     case 4: //init SIM/MODEM
222         printDebugLN(F("Query State of Initialization"));
223         if(sendATcommand("+QINISTAT", "3")==1){

```

```

224     offsetTime=0; state++;
225     printDebugLN(F("READY"));
226     }else{Serial.print(F(".")); delay(100);}
227     clearagsmSerial(); delay(100);
228     offsetTime = millis();
229     break;
230
231     case 5://set audio channel
232     printDebugLN("Audio setup");
233     setAUDIOchannel(20);
234     offsetTime = millis();
235     state++;
236     break;
237
238
239     case 6://let's dial remote
240     if(!offsetTime) offsetTime = millis();
241     if ((millis() - offsetTime) > 5000) restartMODEM();
242
243     printDebugLN("Try to dial the receiptment! After answer, the call can be
244     released hanging up from remote.");
245
246     callStatus =-2;//go to loop and force dial
247     while(callStatus!=0) {
248         if(callStatus < 0) {//no connection, BUSY, ERROR
249             hangup();
250             delay(2000);
251             dial(destinationNumber);
252             printDebugLN("Let's dial receiptment!");
253             printDebugLN("Waiting for remote to answer!");
254         }
255         delay(750);
256         callStatus = getcallStatus();
257     }
258     printDebugLN(F("Answer...wait a while"));
259     delay(2000);//wait a little bit
260
261     while(getcallStatus()==0){//pooling for line status
262         Serial.print(F("."));
263         delay(1000);
264     }
265     printDebugLN(F("\r\nhangup detected"));
266
267     printDebugLN("Call released");
268     delay(5000);//wait a little bit
269
270     clearBUFFD();
271     clearagsmSerial();
272
273     delay(10000);
274     offsetTime = millis();
275     state++;
276     break;
277
278     default:
279     printDebugLN("That's all folks!");
280     delay(100000);
281     restartMODEM();
282     //state=0;
283     break;
284 }

```

285

286 }

287