```
/*
gsm_UTILITIES v 0.921/20171130 - a-gsmII 2.105/b-gsmgnss 2.105
SIM/MODEM/NETWORK/POWER ON/POWER OFF utilities
COPYRIGHT (c) 2014-2017 Dragos Iosub / R&D Software Solutions srl

*****************************IMPORTANT
NOTICE*****************************************
"agsmII_basic_lbr.h", "agsmII_SMS_lbr.ino"
or,
"bgsmgnss_basic_lbr.h", "bgsmgnss_SMS_lbr.ino"
ARE REQUIERED IN ORDER TO RUN THIS EXAMPLE!!!!!!!!!!!!!!!!!!!
Download the "a-gsmII kickstart for Arduino"/"b-gsmgnss kickstart for Arduino" from
here:
https://itbrainpower.net/downloads
Uncompress the archive and copy the files mentined above in the folder
where is this utility, then you can compile this code.

You may like to modify the variables found at line 52
*****************************END of
NOTICE******************************************

You are legaly entitled to use this SOFTWARE ONLY IN CONJUNCTION WITH
a-gsmII/b-gsmgnss DEVICES USAGE. Modifications, derivates and redistribution
of this software must include unmodified this COPYRIGHT NOTICE. You can redistribute
this SOFTWARE and/or modify it under the terms
of this COPYRIGHT NOTICE. Any other usage may be permited only after written notice
of Dragos Iosub / R&D Software Solutions srl.

This SOFTWARE is distributed is provide "AS IS" in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied
warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Dragos Iosub, Bucharest 2017.
http://itbrainpower.net
*/
/*
In order to make the Arduino serial communication (especially for Arduino Uno) with
a-gsmII/b-gsmgnss shield reliable you must
edit C:\Program Files\Arduino\libraries\SoftwareSerial\SoftwareSerial.cpp
comment at line 42
  #define _SS_MAX_RX_BUFF 64 ( will look like: //#define _SS_MAX_RX_BUFF 64 )
and add at next line
  #define _SS_MAX_RX_BUFF 128
You just increased increase the RX buffer size speed for UNO and other snails...
*/
/*
Eventually, you may want to test activateTopSIM() or activateBottomSIM()...just add
code somewhere in the loop(), at the end of one case, like:
case xx:
  ....some previous code
  activateBottomSIM();// or activateTopSIM()...Active SIM value will be stored in
  activeSIM var
  state=0;
  ...in setup() function
Enjoy!
*/
//#define atDebug
//next 2 definition: leave them commented for standard conectivity over Software
serial
//#define usejLader                     //un-comment this if you use micro and nano
GSM 3G adapter for ArduinoNano --Do not use it with a-gsmII/b-gsmgnss!!!!
//#define HARDWARESERIAL                //remove comment to use Serial1 for
```

```
communication on AT MEGA 2560...DUE..

//change next to fit your destination number
char destinationNumber[]="";      //usually phone number with International prefix eg. +40 for Romania

//void activateTopSIM(void){setActiveSIM(0);}//call this function if you want to test/use the a-gsmII/b-gsmgnss primary SIM card
//void activateBottomSIM(void){setActiveSIM(1);};//call this function if you want to test/use the a-gsmII/b-gsmgnss second SIM card

/*do not change under this line! Insteed, make one copy for playing with.*/
#define powerPIN        7//Arduino Digital pin used to power up / power down the modem
#define resetPIN        6//Arduino Digital pin used to reset the modem
#define statusPIN       5//Arduino Digital pin used to monitor if modem is powered

#if (ARDUINO >= 100)
  #include "Arduino.h"
    #if !defined(HARDWARESERIAL)
        #include <SoftwareSerial.h>
    #endif
#else
  #include "WProgram.h"
    #if !defined(HARDWARESERIAL)
        #include <NewSoftSerial.h>
    #endif
#endif

#if defined(HARDWARESERIAL)
    #define BUFFDSIZE 1024
#else
    #if defined(__AVR_ATmega1280__) /*AT MEGA ADK*/|| defined(__AVR_ATmega2560__) /*AT MEGA 2560*/|| defined(__AVR_ATmega32U4__) /*LEONARDO*/
        SoftwareSerial agsmSerial(10,3);  //RX==>10,TX soft==>3...read
        #define BUFFDSIZE 1024
    #else/*UNO*/
        #define UNO_MODE  //Arduino UNO
        #define BUFFDSIZE 200 //240
        #if defined usejLader
            SoftwareSerial agsmSerial(3, 2);  //RX==>3 ,TX soft==>2
        #else
            SoftwareSerial agsmSerial(2, 3);  //RX==>2 ,TX soft==>3
        #endif
    #endif
#endif


#include "agsmII_basic_lbr.h"

#define printDebugLN(x){Serial.println(x);}

int state=0, i=0, powerState = 0;
char ch;
char buffd[256];
//char IMEI[18];
unsigned long offsetTime;
int totalChars = 0;
int ready4SMS = 0;
int ready4Voice = 0;
char readBuffer[200];

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
```

```
          state++;
        i=0;
        res= 0;
        while(res != 1){
          res = sendATcommand("","OK","ERROR",2);
          if (res != 1) {
            if(i++ >= 10) {
              printDebugLN(F("AT err...restarting"));
              restartMODEM();
            }
          }
          delay(500);
        }
        sendATcommand("+IPR=0;&w","OK","ERROR",2);
        delay(2000);
      break;

      case 1:
        clearBUFFD();
        //next some init strings...
        aGsmCMD("AT+QIMODE=0",200);
        aGsmCMD("AT+QINDI=0",200);
        aGsmCMD("AT+QIMUX=0",200);
        aGsmCMD("AT+QIDNSIP=0",200);
        offsetTime=0;
        clearBUFFD();
        state++;
        break;

      case 2:
        printDebugLN(F("try CPIN..."));
        if(!offsetTime) offsetTime = millis();
        if ((millis() - offsetTime) > 20000) restartMODEM();
        if(sendATcommand("+CPIN?","READY")==1){
          offsetTime=0; state++;
          printDebugLN(F("READY"));
        }else{}
        clearagsmSerial(); delay(100);
        offsetTime = millis();
        break;

      case 3:
        if(!offsetTime) offsetTime = millis();
        if ((millis() - offsetTime) > 30000) restartMODEM();

        printDebugLN(F("Query GSM registration?"));
        res = registration(GSM);
        if(res==1){
          offsetTime=0; state++;
          printDebugLN(F("READY, HOME NETWORK"));
        }else if(res==5){
          offsetTime=0; state++;
          printDebugLN(F("READY, ROAMING"));
        }else{
          Serial.print(F("."));
        }
        offsetTime = millis();
      break;

      case 4: //init SIM/MODEM
        printDebugLN(F("Query State of Initialization"));
        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 receipement! After answer, the call can be
          released hanging up from remote.");
244
245         callStatus =-2;//go to loop and force dial
246         while(callStatus!=0) {
247             if(callStatus < 0) {//no connection, BUSY, ERROR
248                 hangup();
249                 delay(2000);
250                 dial(destinationNumber);
251                 printDebugLN("Let's dial receipment!");
252                 printDebugLN("Waiting for remote to answer!");
253             }
254             delay(750);
255             callStatus = getcallStatus();
256         }
257         printDebugLN(F("Answer...wait a while"));
258         delay(2000);//wait a little bit
259
260         while(getcallStatus()==0){//pooling for line status
261           Serial.print(F("."));
262           delay(1000);
263         }
264         printDebugLN(F("\r\nhangup detected"));
265
266         printDebugLN("Call released");
267         delay(5000);//wait a little bit
268
269         clearBUFFD();
270         clearagsmSerial();
271
272         delay(10000);
273         offsetTime = millis();
274         state++;
275       break;
276
277
278       default:
279         printDebugLN("That's all folks!");
280         delay(100000);
281         restartMODEM();
282         //state=0;
283       break;
284     }
```

```
285
286 }
287
```