

```

1  /*
2  DTMF_SEND v 0.921/20171130 - a-gsmII 2.105/b-gsmgnss 2.105 send DTMF example utility
3  COPYRIGHT (c) 2014-2017 Dragos Iosub / R&D Software Solutions srl
4
5  *****IMPORTANT
6  NOTICE*****
7  "agsmII_basic_lbr.h", "agsmII_DTMF_lbr.ino" and "agsmII_basic_lbr.h",
8  "agsmII_DTMF_lbr.ino"
9  or,
10 "bgsmgnss_basic_lbr.h", "bgsmgnss_DTMF_lbr.ino" and "bgsmgnss_basic_lbr.h",
11 "bgsmgnss_DTMF_lbr.ino"
12 ARE REQUIERED IN ORDER TO RUN THIS EXAMPLE!!!!!!!!!!!!!!!!!!!!!!
13
14 Download the "a-gsmII kickstart for Arduino"/"b-gsmgnss kickstart for Arduino" from
15 here:
16 https://itbrainpower.net/downloads
17 Uncompress the archive and copy the files mentined above in the folder
18 where is this utility, then you can compile this code.
19
20 You may want to modify the variables found at lines 45==>49 (use the same values
21 used in DTMF RECEIVE)
22 *****END of
23 NOTICE*****
24
25 You are legally entitled to use this SOFTWARE ONLY IN CONJUNCTION WITH
26 a-gsmII/b-gsmgnss DEVICES USAGE. Modifications, derivate and redistribution
27 of this software must include unmodified this COPYRIGHT NOTICE. You can redistribute
28 this SOFTWARE and/or modify it under the terms
29 of this COPYRIGHT NOTICE. Any other usage may be permitted only after written notice
30 of Dragos Iosub / R&D Software Solutions srl.
31
32 This SOFTWARE is distributed is provide "AS IS" in the hope that it will be useful,
33 but WITHOUT ANY WARRANTY; without even the implied
34 warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
35
36 Dragos Iosub, Bucharest 2017.
37 http://itbrainpower.net
38 */
39 /*
40 edit C:\Program Files\Arduino\libraries\SoftwareSerial\SoftwareSerial.h
41 In order to make the Arduino serial communication (especially for Arduino Uno) with
42 a-gsmII/b-gsmgnss shield reliable you must
43 comment at line 42
44 #define _SS_MAX_RX_BUFFER 64 ( will look like: //define _SS_MAX_RX_BUFFER 64 )
45 and add at next line
46 #define _SS_MAX_RX_BUFFER 128
47 You just increased increase the RX buffer size speed for UNO and other snails...
48 */
49
50 //next 2 definition: leave them commented for standard conectivity over Software
51 serial
52 //define usejLader //un-comment this if you use micro and nano
53 GSM 3G adapter for ArduinoNano --Do not use it with a-gsmII/b-gsmgnss!!!!
54 //define HARDWARESERIAL //remove comment to use Serial1 for
55 communication on AT MEGA 2560...DUE..
56
57 //define atDebug //uncomment this to debug serial communication with
58 a-gsmII/b-gsmgnss
59
60 int DTMFlenght=100; //DTMF lenght in miliseconds - 90-100ms best value for manual
61 int DTMFpause=100; //pause lenght between DTMF in miliseconds - 90-100ms best
62 value for manual

```

```

47 //change next line to fit your destination number!
48 char destinationNumber[]=""; //usually phone number with International prefix
   eg. *40 for Romania
49 char message[]="ABCD0123456789*#***"; //last 3 chars -*** are used as
   terminator in DTMF_RECEIVE (take a look at DTMF_RECEIVE code)
50
51
52 /*do not change under this line! Instead, make one copy for playing with.*/
53 #define powerPIN 7//Arduino Digital pin used to power up / power down the modem
54 #define resetPIN 6//Arduino Digital pin used to reset the modem
55 #define statusPIN 5//Arduino Digital pin used to monitor if modem is powered
56
57 #if (ARDUINO >= 100)
58 #include "Arduino.h"
59 #if !defined(HARDWARESERIAL)
60 #include <SoftwareSerial.h>
61 #endif
62 #else
63 #include "WProgram.h"
64 #if !defined(HARDWARESERIAL)
65 #include <NewSoftSerial.h>
66 #endif
67 #endif
68
69 #if defined(HARDWARESERIAL)
70 #define BUFFDSIZE 1024
71 #else
72 #if defined(__AVR_ATmega1280__) /*AT MEGA ADK*/|| defined(__AVR_ATmega2560__)
   /*AT MEGA 2560*/|| defined(__AVR_ATmega32U4__) /*LEONARDO*/
73 SoftwareSerial agsmSerial(10,3); //RX==>10,TX soft==>3...read
74 #define BUFFDSIZE 1024
75 #else/*UNO*/
76 #define UNO_MODE //Arduino UNO
77 #define BUFFDSIZE 200 //240
78 #if defined usejLader
79 SoftwareSerial agsmSerial(3, 2); //RX==>3 ,TX soft==>2
80 #else
81 SoftwareSerial agsmSerial(2, 3); //RX==>2 ,TX soft==>3
82 #endif
83 #endif
84 #endif
85
86
87 #include "agsmII_DTMF_lbr.h"
88
89 #define printDebugLN(x){Serial.println(x);}
90
91 int state=0, i=0, powerState = 0;
92 char ch;
93 char buffd[256];
94 //char IMEI[18];
95 unsigned long offsetTime;
96 int totalChars = 0;
97 int ready4SMS = 0;
98 int ready4Voice = 0;
99 char readBuffer[200];
100
101 void setup(){
102 agsmSerial.begin(9600);
103 Serial.begin(57600);
104 clearagsmSerial();
105 clearSerial();

```

```

106     delay(10);
107
108     modemHWSetup(); //configure Arduino IN and OUT to be
    used with modem
109
110     Serial.flush();
111     agsmSerial.flush();
112     delay(1000);
113     Serial.println(F("a-gsmII/b-gsmgnss DTMF SEND example"));
114     Serial.flush();
115
116     if(strlen(destinationNumber)<1) {
117         Serial.print(F("destinationNumber not initialized. Edit DTMF_SEND_SS.ino and
    set the destinationNumber(line 38) with your phone number.\r\n\r\nNow the
    program will stop.));
118         delay(1000);
119         exit(0);
120     }
121
122     Serial.println(F("seat back and relax until a-gsmII/b-gsmgnss is ready"));
123     delay(100);
124
125     powerOnModem();
126
127     clearBUFFD();
128     while(strlen(buffd)<1) {
129         getIMEI();
130         delay(500);
131     }
132
133     ready4SMS = 0;
134     ready4Voice = 0;
135
136     Serial.println(F("a-gsmII/b-gsmgnss ready.. let's run the example"));
137     Serial.print(F("a-gsmII/b-gsmgnss IMEI: ")); Serial.flush();
138     Serial.println(buffd); Serial.flush();
139     //setAUDIOchannel(20);
140     delay(1000);
141 }
142
143 void loop(){
144     //char readFileBuffer[128];
145     int callStatus;
146     int res;
147     int count=0;
148
149     switch(state){
150     case 0://check modem status
151         if(!getModemState()) restartMODEM();
152         else
153             state++;
154         i=0;
155         res= 0;
156         while(res != 1){
157             res = sendATcommand("", "OK", "ERROR", 2);
158             if (res != 1) {
159                 if(i++ >= 10) {
160                     printDebugLN(F("AT err...restarting"));
161                     restartMODEM();
162                 }
163             }
164             delay(500);

```

```

165     }
166     sendATcommand("+IPR=0;&w","OK","ERROR",2);
167     delay(2000);
168     break;
169
170     case 1:
171         clearBUFFD();
172         //next some init strings...
173         aGsmCMD("AT+QIMODE=0",200);
174         aGsmCMD("AT+QINDI=0",200);
175         aGsmCMD("AT+QIMUX=0",200);
176         aGsmCMD("AT+QIDNSIP=0",200);
177         offsetTime=0;
178         clearBUFFD();
179         state++;
180         break;
181
182     case 2:
183         printDebugLN(F("try CPIN..."));
184         if(!offsetTime) offsetTime = millis();
185         if ((millis() - offsetTime) > 20000) restartMODEM();
186         if(sendATcommand("+CPIN?", "READY")==1){
187             offsetTime=0; state++;
188             printDebugLN(F("READY"));
189         }else{}
190         clearagsmSerial(); delay(100);
191         offsetTime = millis();
192         break;
193
194     case 3:
195         if(!offsetTime) offsetTime = millis();
196         if ((millis() - offsetTime) > 30000) restartMODEM();
197
198         printDebugLN(F("Query GSM registration?"));
199         res = registration(GSM);
200         if(res==1){
201             offsetTime=0; state++;
202             printDebugLN(F("READY, HOME NETWORK"));
203         }else if(res==5){
204             offsetTime=0; state++;
205             printDebugLN(F("READY, ROAMING"));
206         }else{
207             Serial.print(F("."));
208         }
209         offsetTime = millis();
210         break;
211
212     case 4: //init SIM/MODEM
213         printDebugLN(F("Query State of Initialization"));
214         if(sendATcommand("+QINISTAT","3")==1){
215             offsetTime=0; state++;
216             printDebugLN(F("READY"));
217         }else{Serial.print(F(".")); delay(100);}
218         clearagsmSerial(); delay(100);
219         offsetTime = millis();
220         break;
221
222     case 5://Modem full initialised?
223         if(!offsetTime) offsetTime = millis();
224         if ((millis() - offsetTime) > 5000) restartMODEM();
225         clearBUFFD();
226         clearagsmSerial();

```

```

227
228
229     printDebugLN("It is Modem full initialised?"); delay(100);
230     setupMODEMforDTMFSusage();
231     delay(10000);
232     offsetTime = millis();
233     state++;
234     break;
235
236
237     case 6://let's send DTMF to the destination receiptment
238         if(!offsetTime) offsetTime = millis();
239         if ((millis() - offsetTime) > 5000) restartMODEM();
240
241
242         printDebugLN(F("Let's dial the receiptment!"));
243         //memset(readBuffer,0x00,sizeof(readBuffer));
244         //sprintf(readBuffer,"D%s;",destinationNumber);//prepare dial command
245         //printDebugLN(readBuffer);
246
247         printDebugLN(F("Waiting for remote to answer!"));
248
249         callStatus =-2;//go into loop and force dial
250         while(callStatus!=0) {
251             if(callStatus < 0) {//no connection, BUSY, ERROR
252                 hangup();
253                 delay(2000);
254                 dial(destinationNumber);
255             }
256             delay(500);
257             callStatus = getcallStatus();
258         }
259         printDebugLN(F("Answer...wait a while"));
260         delay(2000);//wait a little bit
261
262         while(getcallStatus()==0){//send DTMF, 20sec pause, until line is no connected
263             sendDTMF(message);
264             printDebugLN(F("DTMF send, repeat in 5sec while hangup is detected"));
265             delay(5000);
266         }
267         printDebugLN(F("hangup detected"));
268
269         delay(10000);
270
271         printDebugLN(F("That's all folks!"));
272
273         delay(10000);
274         offsetTime = millis();
275         state++;
276         break;
277
278     default:
279         //restartMODEM();
280         delay(1000000);
281         //state=0;
282     break;
283 }
284
285 }
286

```