

```

1  /*
2  GPRS_HTTP v 0.921/20171130 - a-gsmII 2.105/b-gsmgnss 2.105 HTTP client over GPRS
example utility
3  COPYRIGHT (c) 2014-2017 Dragos Iosub / R&D Software Solutions srl
4
5  *****IMPORTANT
NOTICE*****
6  "agsmII_basic_lbr.h", "agsmII_IP_lbr.ino" and "agsmII_basic_lbr.h",
"agsmII_IP_lbr.ino" and user_GPRS_HTTP_PARS.h
7  or,
8  "bgsmgnss_basic_lbr.h", "bgsmgnss_IP_lbr.ino" and "bgsmgnss_basic_lbr.h",
"bgsmgnss_IP_lbr.ino" and user_GPRS_HTTP_PARS.h
9  ARE REQUIERED IN ORDER TO RUN THIS EXAMPLE!!!!!!!!!!!!!!!!!!!!!!
10
11  Download the "a-gsmII kickstart for Arduino"/"b-gsmgnss kickstart for Arduino" from
here:
12  https://itbrainpower.net/downloads
13  Uncompress the archive and copy the files mentined above in the folder
14  where is this utility, then you can compile this code.
15
16  You may want to modify the variables found in first 3 parameters
"user_GPRS_HTTP_PARS.h" - use the GPRS related parameters
17  provided by your GSM operator (default ORANGE RO network)
18
19  For SSL encrypted (https) data transfer in just comment the line 125 and remove
comment from the line 126.
20  *****END of
NOTICE*****
21
22  You are legaly entitled to use this SOFTWARE ONLY IN CONJUNCTION WITH
a-gsmII/b-gsmgnss DEVICES USAGE. Modifications, derivates and redistribution
23  of this software must include unmodified this COPYRIGHT NOTICE. You can redistribute
this SOFTWARE and/or modify it under the terms
24  of this COPYRIGHT NOTICE. Any other usage may be permitted only after written notice
of Dragos Iosub / R&D Software Solutions srl.
25
26  This SOFTWARE is distributed is provide "AS IS" in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied
27  warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
28
29  Dragos Iosub, Bucharest 2017.
30  http://itbrainpower.net
31  */
32  /*
33  Server php script: simple echoes the $_GET and $_POST variable received ---> just
to be displayed on the Arduino ;)
34  */
35
36  //#define atDebug //uncomment this to debug serial communication with
a-gsmII/b-gsmgnss
37
38  //next 2 definition: leave them commented for standard conectivity over Software
serial
39  //#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!!!!
40  //#define HARDWARESERIAL //remove comment to use Serial1 for
communication on AT MEGA 2560...DUE..
41
42  /*do not change under this line! Instead, make one copy for playing with.*/
43  #include "agsmII_IP_lbr.h"
44
45  #define powerPIN 7//Arduino Digital pin used to power up / power down the modem

```

```

46 #define resetPIN          6//Arduino Digital pin used to reset the modem
47 #define statusPIN        5//Arduino Digital pin used to monitor if modem is powered
48
49 #if (ARDUINO >= 100)
50     #include "Arduino.h"
51     #if !defined(HARDWARESERIAL)
52         #include <SoftwareSerial.h>
53     #endif
54 #else
55     #include "WProgram.h"
56     #if !defined(HARDWARESERIAL)
57         #include <NewSoftSerial.h>
58     #endif
59 #endif
60
61 #if defined(HARDWARESERIAL)
62     #define BUFFDSIZE 1024
63 #else
64     #if defined(__AVR_ATmega1280__) /*AT MEGA ADK*/|| defined(__AVR_ATmega2560__)
65         /*AT MEGA 2560*/|| defined(__AVR_ATmega32U4__) /*LEONARDO*/
66         SoftwareSerial agsmSerial(10,3); //RX==>10,TX soft==>3...read
67         #define BUFFDSIZE 1024
68     #else/*UNO*/
69         #define UNO_MODE //Arduino UNO
70         #define BUFFDSIZE 200 //240
71         #if defined usejLader
72             SoftwareSerial agsmSerial(3, 2); //RX==>3 ,TX soft==>2
73         #else
74             SoftwareSerial agsmSerial(2, 3); //RX==>2 ,TX soft==>3
75         #endif
76     #endif
77 #endif
78
79
80 #define printDebugLN(x){Serial.println(x);}
81
82 int state=0, i=0, powerState = 0;
83 char ch;
84 char readBuffer[200];
85 char buffd[BUFFDSIZE];
86 int ready4SMS = 0;
87 int ready4Voice = 0;
88
89
90 void setup(){
91     agsmSerial.begin(9600);
92     Serial.begin(57600);
93     clearagsmSerial();
94     clearSerial();
95     delay(10);
96
97     modemHWSetup(); //configure Arduino IN and OUT to be
98                     used with modem
99
100    Serial.flush();
101    agsmSerial.flush();
102    delay(1000);
103    Serial.println(F("a-gsmII/b-gsmgnss HTTP GET and POST upload variables and
download server respose example\r\n"));
104    Serial.println(F("***GPRS plan needed & check the GPRS settings in
\"user_GPRS_HTTP_PARS.h\"***\r\n"));

```

```

104     Serial.flush();
105
106     Serial.println(F("seat back and relax until a-gsmII/b-gsmgnss is ready"));
107     delay(100);
108
109     powerOnModem();
110
111     clearBUFFD();
112     while(strlen(buffd)<1){
113         getIMEI();
114         delay(500);
115     }
116
117     ready4SMS = 0;
118     ready4Voice = 0;
119
120     Serial.println(F("a-gsmII/b-gsmgnss ready.. let's run the example"));
121     Serial.print(F("a-gsmII/b-gsmgnss IMEI: ")); Serial.flush();
122     Serial.println(buffd); Serial.flush();
123     //setAUDIOchannel(20);
124     delay(1000);
125     setSSLMODE(SSLDISABLED);//http mode
126     //setSSLMODE(SSLENABLED);//https mode - SSL ENABLED mode it is supported only by
newest a-gsmII/b-gsmgnss with part numbers like (AGSM2064#xSyAP-SSL)
127 }
128
129
130
131 void loop(){
132     int counter=0;
133     while (counter<3){
134         printDebugLN("try send a=112, test=33020128, Data=322 data via GET method");
135         //HTTP_REQUEST("a=112&IMEI=33020128&Data=322",GET);//use GET method without
timeout
136         PROCESS_HTTP_REQUEST("a=112&test=33020128&Data=322",GET,
HTTP_PROCESSING_GENERAL_TIMEOUT);//use GET method with timeout
137         /*Serial.print(F("process takes: "));
138         Serial.print(millis() - HTTP_STARTTIME);
139         Serial.println(F("msec"));*/
140         delay(2000);
141         printDebugLN("try send a=112, test=33020128, Data=322 data via POST method");
142         //HTTP_REQUEST("a=112&IMEI=33020128&Data=322",POST);//use POST method
without timeout
143         PROCESS_HTTP_REQUEST("a=112&test=33020128&Data=322",POST,
HTTP_PROCESSING_GENERAL_TIMEOUT);//use POST method with timeout
144         delay(2000);
145         /*Serial.print(F("process takes: "));
146         Serial.print(millis() - HTTP_STARTTIME);
147         Serial.println(F("msec"));*/
148         counter++;
149     }
150
151     printDebugLN(F("That's all folks!"));
152     delay(10000);
153     exit(0);
154 }
155

```