```
/*
GPRS_HTTP v 0.921/20160707 - a-gsm 2.064 HTTP client over GPRS example utility
COPYRIGHT (c) 2014-2016 Dragos Iosub / R&D Software Solutions srl

*****************************IMPORTANT NOTICE*****************************************
"agsm_basic_lbr.h", "agsm_IP_lbr.ino" and "agsm_basic_lbr.h", "agsm_IP_lbr.ino" and
user_GPRS_HTTP_PARS.h ARE REQUIERED IN ORDER TO RUN THIS EXAMPLE!!!!!!!!!!!!!!!!!!!
Download the "a-gsm kickstart for Arduino" from the itbrainpower.net download section.
Uncompress the archive and copy the files mentined above in the folder
where is this utility, then you can compile this code.

You may want to modify the first 3 parameters (GPRS related) from user_GPRS_HTTP_PARS.h
YOU MUST USE SIM CARD with valid GPRS data plan!
*****************************END of NOTICE*****************************************

You are legaly entitled to use this SOFTWARE ONLY IN CONJUNCTION WITH a-gsm 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 2016.
http://itbrainpower.net
*/
/*
  Server php script: simple echoes the $_GET and $_POST variable received ---> just to be
  displayed on the Arduino ;)
*/

//#define atDebug //uncomment this to debug serial communication with a-gsm

//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-gsm!!!!
//#define HARDWARESERIAL                //remove comment to use Serial1 for communication on
AT MEGA 2560...DUE..

/*do not change under this line! Insteed, make one copy for playing with.*/
//#include "agsm_basic_lbr.h"
#include "agsm_IP_lbr.h"

#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



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

int state=0, i=0, powerState = 0;
char ch;
char readBuffer[200];
char buffd[BUFFDSIZE];
int ready4SMS = 0;
int ready4Voice = 0;


void setup(){
    agsmSerial.begin(9600);
    Serial.begin(57600);
    clearagsmSerial();
    clearSerial();
    delay(10);

    modemHWSetup();                          //configure Arduino IN and OUT to be used with
    modem

    Serial.flush();
    agsmSerial.flush();
    delay(1000);
    Serial.println(F("a-gsm HTTP GET and POST upload variables and download server respose
    example\r\n"));
    Serial.println(F("***GPRS plan needed & check the GPRS settings in
    \"user_GPRS_HTTP_PARS.h\"***\r\n"));
    Serial.flush();

    Serial.println(F("sit back and relax until a-gsm is ready"));
    delay(100);
```

```
    powerOnModem();

    clearBUFFD();
    while(strlen(buffd)<1){
        getIMEI();
        delay(500);
    }

    ready4SMS = 0;
    ready4Voice = 0;

    Serial.println(F("a-gsm ready.. let's run the example"));
    Serial.print(F("a-gsm IMEI: ")); Serial.flush();
    Serial.println(buffd); Serial.flush();
    //setAUDIOchannel(20);
    delay(1000);
    setSSLMODE(SSLDISABLED);//http mode
    //setSSLMODE(SSLENABLED);//https mode - SSL ENABLED mode it is supported only by newest
    a-gsm with part numbers like (AGSM2064#xSyAP-SSL)
}



void loop(){
    int counter=0;
    while (counter<3){
        printDebugLN("try send  a=112, test=33020128, Data=322 data via GET method");
        //HTTP_REQUEST("a=112&IMEI=33020128&Data=322",GET);//use GET method without timeout
        PROCESS_HTTP_REQUEST("a=112&test=33020128&Data=322",GET,
        HTTP_PROCESSING_GENERAL_TIMEOUT);//use GET method with timeout
        /*Serial.print(F("process takes: "));
        Serial.print(millis() - HTTP_STARTTIME);
        Serial.println(F("msec"));*/
        delay(2000);
        printDebugLN("try send a=112, test=33020128, Data=322 data via POST method");
        //HTTP_REQUEST("a=112&IMEI=33020128&Data=322",POST);//use POST method without timeout
        PROCESS_HTTP_REQUEST("a=112&test=33020128&Data=322",POST,
        HTTP_PROCESSING_GENERAL_TIMEOUT);//use POST method with timeout
        delay(2000);
        /*Serial.print(F("process takes: "));
        Serial.print(millis() - HTTP_STARTTIME);
        Serial.println(F("msec"));*/
        counter++;
    }

    printDebugLN(F("That's all folks!"));
    delay(10000);
    exit(0);
}
```