

```

#include <Arduino.h>
#include "ESP8266WiFi.h"
#include <ESP8266HTTPClient.h>
#include <SoftwareSerial.h> // SoftwareSerial must be included because the library depends on it

/* projet Smartpoker
   version : 1.0
   date : 21/11/2018
   redacteur : slatkin
   website : smartpoker.jimbo.com/esp8266/Arduino
   smartpoker.fr
   this programm allow to a ESP01s to connect to a AP, and send the information (2 RFID reader or
   Keypad) to the website smartpoker.fr
   Trame HTTP envoyé au serveur :
   http://smartpoker.livehost.fr/smartpoker/website/admin/srv/srv.php?
   D=1&Em=Macaddress&Tag=xxxxxx
*/

//WiFi
const char* ssid = "My ASUS crosne";
const char* password = "bureau01?";
const char* host = "smartpoker.livehost.fr";
unsigned long delaiwatchdog = 300000; // Fréquence du watchdog - Watchdog frequency
unsigned long current;
unsigned long previous;
String mac_address="";

int port=80;
HTTPClient http;
String payload="";
int ret=0; // recherche d'une sous chaine dans Payload
int cas=0;

/**/Gestion des échanges avec l'arduino **/
char inChar;
String inputstring = ""; // a String to hold incoming data
bool stringComplete = false; // whether the string is complete
String cmdtoarduino="";

void setup()
{
  Serial.begin(115200);

//connexion au reseau WiFi
  // Set WiFi to station mode and disconnect from an AP if it was previously connected
  /* Explicitly set the ESP8266 to be a WiFi-client, otherwise, it by default,
   would try to act as both a client and an access-point and could cause
   network-issues with your other WiFi-devices on your WiFi-network. */
  WiFi.mode(WIFI_STA);

```

```

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}
// Fin de l'init
delay(3000);

  fnwatchdog();

  //commandes vers l'arduino
  Serial.println("cmd=00"); //ecran 0
  Serial.println("cmd=05"); //efface ecran
  Serial.println("cmd=06"); //textesize=1;
  Serial.println("cmd=21?Initialisation"); // ligne1
  Serial.println("cmd22=WiFi connected");
  Serial.print("cmd=23?Mac Address"); // ligne2
  Serial.print("cmd=24?"); // ligne3
  Serial.println(WiFi.macAddress());
  Serial.print("cmd=25?Ip Address"); // ligne4
  Serial.print("cmd=26?"); // ligne5
  Serial.println(WiFi.localIP().toString());
  Serial.println("cmd=01"); //ecran 0
  Serial.println("cmd=05"); //efface ecran
  previous=millis();
}

void loop()
{
  // Watchdog reccurent
  current=millis();
  if (current>=(previous+delaiwatchdog))
  {
    fnwatchdog();
    previous=millis();

  }

  // lecture du port serie
  while (Serial.available())
  {
    // get the new byte:
    char inChar = (char)Serial.read();

    // add it to the inputString:
    inputstring += inChar;
    // if the incoming character is a newline, set a flag so the main loop can
    // do something about it:
    if (inChar == '\n')
    {

```

```

    stringComplete = true;
    Serial.println("Fin saisie Serial");
    decode_cmd(inputstring);
    inputstring="";
  } //end if Inchar
} //endwhile serial.available
} //end loop

```

```

int decode_cmd(String cmdfromarduino)

```

```

{
  int nbcар=0;
  bool controle_format=false;
  bool envoi_payload=false;
  String payload2="";

```

```

  bool debug=true;

```

```

  /*structure : cmd=xx?yyyyyy

```

```

    tag= :mot clé de longueur fixe , 4 caracteres

```

```

    xxx : est un commande, on peut la laisser au format string, mais 3 caracteres uniquement
    cela correspond aux codes de la table Tcontroles

```

```

    &data= : mot clé , c'est un separateur sur un caractere

```

```

    yyyyy : données variables liées à la commande

```

```

  */

```

```

  /* Parametrage du Moniteur Serie :

```

```

  * 115200 baud

```

```

  * Nouvelle ligne

```

```

  * important ; le rajout de /n en fin de ligne evite de creer une boucle sur le port COM

```

```

  */

```

```

  if (debug)

```

```

  {

```

```

    Serial.print ("cmdfromarduino: ");

```

```

    Serial.println(cmdfromarduino);

```

```

  }

```

```

  /*controle du format de la trame*/

```

```

  nbcар = cmdfromarduino.length()-1; //j'enleve le caractere /n du compteur

```

```

  if (debug)

```

```

  {

```

```

    Serial.print ("Nb car : ");

```

```

    Serial.println(nbcар);

```

```

  }

```

```

  if (nbcар <=8) //if faut rajouter le caractere /n dans

```

```

  {

```

```

    Serial.println("Nombre de caracteres insuffisant");

```

```

    controle_format=false;

```

```

  }

```

```

  /*recherche cmd=, en premiere position */

```

```

  if (cmdfromarduino.startsWith("Tag=",1))

```

```

  {

```

```

Serial.println("mot clé tag OK");
controle_format=true;
}
cmdfromarduino.trim();
String url = "/smartpoker/website/admin/srv/srv.php/D=0&Em=";
url += WiFi.macAddress();
url += "&";
url += cmdfromarduino;
Serial.print("connecting to ");
Serial.println(host);
Serial.print("Requesting URL: ");
Serial.println(url);
http.begin(host,port,url);
int httpCode = http.GET();
if (httpCode)
{
  if (httpCode == 200) {
    payload = http.getString();
    //Serial.println(payload);
    int debut = payload.indexOf("Payload :");
    Serial.println("cmd=01"); //ecran 01
    Serial.println("cmd=05"); //efface ecran
    Serial.println("cmd=06"); //textesize=1;
    Serial.print("cmd=10?"); //Status
    Serial.println(payload.substring(debut,payload.length()));
  }
}
http.end();
/* WiFi.disconnect();*/
/* Analyse du payload */

/* envoi du Payload vers l'arduino*/
if (envoi_payload=true)
{
  envoi_cmd(payload2);
  envoi_payload=false;
} //end if envoi_payload
} //end decode_cmd

int envoi_cmd(String cmdtoarduino)
{
  Serial.println(cmdtoarduino);
}

int fnwatchdog()
{
  Serial.println("Envoi d'une trame watchdog....");
  String url = "/smartpoker/website/watchdog/watchdog.php?mac=";
  url += WiFi.macAddress();
  url += "&ip=";
  url += WiFi.localIP().toString();
}

```

```
Serial.print("connecting to ");
Serial.println(host);
Serial.print("Requesting URL: ");
Serial.println(url);

http.begin(host,port,url);
int httpCode = http.GET();
if (httpCode)
{
  if (httpCode == 200) {
    payload = http.getString();
    Serial.println(payload);
    Serial.println("Fin payload");
  }
}
http.end();
}
```