



## Trabalho 1 - Comunicações Cliente/Servidor

### Introdução:

No âmbito da cadeira de Sistemas de Comunicação Móvel foi-nos proposto a implementação de um sistema de comunicação entre cliente/servidor usando 2 arduinos esp8266.

### Detalhes de implementação:

Como ponto de partida foi usado o exemplo “WiFiManualWebServer.ino” para o servidor e o exemplo “WiFiClientBasic.ino” para o cliente. A primeira alteração feita foi no *SSID* e *Password* que, por sua vez, foram modificados de forma a que os arduinos pudessem conectar-se a uma rede *WiFi* (neste caso, um *hotspot*), sendo que do lado do cliente foi ainda adicionado o ip referente ao servidor.

De seguida, foi adicionado 1 ciclo *do while* nos códigos de ambas as partes que terá como função suportar a comunicação até que um dos lados envie o comando “STOP” através da consola.

### Código implementado:

```
//the only thing added/changed
String req = "";
String string = "";
int variant_num;
String variant = "";

do{

    //if there exists something to read from client side then
    if(client.available() != 0){
        req = client.readStringUntil('\n');
        //depending on the "command" received
        //ON - turn on my led
        //OFF - turn off my led
        //WIFI - retrieve info about my wifi connection (SSID, channel, RSSI)
        if(req == "ON"){
            digitalWrite(LED_BUILTIN, 0);
            client.println("Other LED turned on");
        }
        else if(req == "OFF"){
            digitalWrite(LED_BUILTIN, 1);
            client.println("Other LED turned off");
        }
        else if(req == "WIFI"){
            variant_num = WiFi.getPhyMode();

            if(variant_num == WIFI_PHY_MODE_11B){
                variant = "802.11 b";
            }
            else if(variant_num == WIFI_PHY_MODE_11G){
                variant = "802.11 g";
            }
            else if(variant_num == WIFI_PHY_MODE_11N){
                variant = "802.11 n";
            }
            else{
                variant = "802.11 (other version)";
            }

            client.println("Name of the network: " + WiFi.SSID() + "\nChannel used: " + WiFi.channel() + "\nRSSI of the network: " + WiFi.RSSI() + "dbm\n802.11 variant: " + variant);
        }
        else if(req == "STOP"){
            string = "STOP\n";
        }
        else{
            Serial.println(req);
        }
    }
    //if there exists something to read from serial monitor
    else if(Serial.available() != 0){
        string = Serial.readString();
        client.print(string);
    }

    req = "";
}while (string != "STOP\n");

Serial.println("Closing connection...");
```



O ciclo é igual no código dos dois arduinos (cliente/ servidor). No entanto, com o intuito de detetar um input na consola foi usada a função `Serial.available()` que permite contar quantos caracteres foram escritos na mesma. Por sua vez, para verificar se o outro arduino enviou algo é usada a função `client.available()` que novamente visa contar quantos caracteres foram escritos, mas desta vez pelo arduino contrário.

Foram definidos alguns comandos para que a implementação das funcionalidades pedidas fosse possível:

- ON - Permite ligar o LED do outro arduino;
- OFF - Permite desligar o LED do outro arduino;
- WIFI - Retorna a informação da rede *WiFi* a que o outro arduino se encontra ligado (SSID, Canal, RSSI, 802.11 variant);
- STOP - Permite terminar a ligação entre o cliente e o servidor.

#### **Problemas com que nos deparamos:**

- Inicialmente, como não conhecíamos a função `Serial.available()` tentámos utilizar outras estratégias que apresentassem essa mesma funcionalidade. No entanto, estas não demonstraram ser as melhores, vindo mesmo a causar alguns problemas, entre os quais o envio de uma string vazia sem que tivesse havido qualquer input por parte do utilizador.
- Para que fosse possível terminar a ligação entre o cliente e o servidor começámos por efetuar um *break* assim que fosse introduzido ou recebido o input STOP na consola. No entanto, verificamos que esta abordagem não era a mais correta, pois a ligação entre ambos não era completamente fechada. Assim, optamos por acrescentar no lado do cliente o método `client.stop()`.



## Código do servidor:

```
//Inês Martins Marçal N°: 2019215917
//João Carlos Borges Silva N°: 2029216753

#include <ESP8266WiFi.h>

#ifndef STASSID
#define STASSID "GalaxyA21s24AA" //"joaophone"
#define STAPSK "zuyu4125" //"123456789"
#endif

const char* ssid = STASSID;
const char* password = STAPSK;

// Create an instance of the server
// specify the port to listen on as an argument
WiFiServer server(25);

void setup() {
  Serial.begin(9600);

  // prepare LED
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, 0);

  // Connect to WiFi network
  Serial.println();
  Serial.println();
  Serial.print(F("Connecting to "));
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(F("."));
  }
  Serial.println();
  Serial.println(F("WiFi connected"));

  // Start the server
  server.begin();
  Serial.println(F("Server started"));

  // Print the IP address
  Serial.println(WiFi.localIP());
}

void loop() {
  // Check if a client has connected
  WiFiClient client = server.available();
  if (!client) {
    return;
  }
  Serial.println(F("new client"));

  client.setTimeout(5000); // default is 1000

  // Match the request
  /*int val;
  if (req.indexOf(F("/gpio/0")) != -1) {
    val = 0;
  } else if (req.indexOf(F("/gpio/1")) != -1) {
    val = 1;
  } else {
    Serial.println(F("invalid request"));
    val = digitalRead(LED_BUILTIN);
  }

  // Set LED according to the request
  digitalWrite(LED_BUILTIN, val);*/

  // read/ignore the rest of the request
  // do not client.flush(): it is for output only, see below

  //the only thing added/changed
  String req = "";
  String string = "";
  int variant_num;
  String variant = "";

  do{

    //if there exists something to read from client side then
    if(client.available() != 0){
      req = client.readStringUntil('\n');
      //depending on the "command" received
      //ON - turn on my led
      //OFF - turn off my led
      //WIFI - retrieve info about my wifi connection (SSID, channel, RSSI)
      if(req == "ON"){
```



Universidade de Coimbra  
Faculdade de Ciências e Tecnologias da Universidade de Coimbra  
Mestrado em Segurança Informática

```
digitalWrite(LED_BUILTIN, 0);
client.println("Other LED turned on");
}
else if(req == "OFF"){
digitalWrite(LED_BUILTIN, 1);
client.println("Other LED turned off");
}
else if(req == "WIFI"){
variant_num = WiFi.getPhyMode();

if(variant_num == WIFI_PHY_MODE_11B){
variant = "802.11 b";
}
else if(variant_num == WIFI_PHY_MODE_11G){
variant = "802.11 g";
}
else if(variant_num == WIFI_PHY_MODE_11N){
variant = "802.11 n";
}
else{
variant = "802.11 (other version)";
}

client.println("Name of the network: " + WiFi.SSID() + "\nChannel used: " + WiFi.channel()
+ "\nRSSI of the network: " + WiFi.RSSI() + "dbm\n802.11 variant: " + variant);
}
else if(req == "STOP"){
string = "STOP\n";
}
else{
Serial.println(req);
}
}
//if there exists something to read from serial monitor
else if(Serial.available() != 0){
string = Serial.readString();
client.print(string);
}
}

req = "";
}while (string != "STOP\n");

Serial.println("Closing connection...");
}
```



## Código do cliente:

```
//Inês Martins Marçal N°: 2019215917
//João Carlos Borges Silva N°: 201916753

#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>

#ifdef STASSID
#define STASSID "GalaxyA21s24AA" //nome do hotspot
#define STAPSK "zuyu4125" //password do hotspot
#endif

const char* ssid = STASSID;
const char* password = STAPSK;

const char* host = "192.168.250.242";
const uint16_t port = 25;

ESP8266WiFiMulti WiFiMulti;

void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
  // We start by connecting to a WiFi network
  WiFi.mode(WIFI_STA);
  WiFiMulti.addAP(ssid, password);

  Serial.println();
  Serial.println();
  Serial.print("Wait for WiFi... ");

  while (WiFiMulti.run() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  delay(500);
}

void loop() {
  Serial.print("connecting to ");
  Serial.print(host);
  Serial.print(":");
  Serial.println(port);

  // Use WiFiClient class to create TCP connections
  WiFiClient client;

  if (!client.connect(host, port)) {
    Serial.println("connection failed");
    Serial.println("wait 5 sec...");
    delay(5000);
    return;
  }

  //the only thing added/changed
  String req = "";
  String string = "";
  int variant_num;
  String variant = "";

  do{

    //if there exists something to read from serial monitor
    if(Serial.available() != 0){
      string = Serial.readString();
      client.print(string);
    }

    //if there exists something to read from client side then
    else if(client.available() != 0){
      req = client.readStringUntil('\n');
      //depeding on the "command" received
      //ON - turn on my led
      //OFF - turn off my led
      //WIFI - retrieve info about my wifi connection (SSID, channel, RSSI, 802.11 variant)
      if(req == "ON"){
        Serial.println(req+"|");
        digitalWrite(LED_BUILTIN, 0);
        client.println("Other LED turned on");
      }
      else if(req == "OFF"){
        digitalWrite(LED_BUILTIN, 1);
        client.println("Other LED turned off");
      }
    }
  }
```



Universidade de Coimbra  
Faculdade de Ciências e Tecnologias da Universidade de Coimbra  
Mestrado em Segurança Informática

```
else if(req == "WIFI"){
    Serial.println(req+"|");
    variant_num = WiFi.getPhyMode();

    if(variant_num == WIFI_PHY_MODE_11B){
        variant = "802.11 b";
    }
    else if(variant_num == WIFI_PHY_MODE_11G){
        variant = "802.11 g";
    }
    else if(variant_num == WIFI_PHY_MODE_11N){
        variant = "802.11 n";
    }
    else{
        variant = "802.11 (other version)";
    }
    client.println("Name of the network: " + WiFi.SSID() + "\nChannel used: " + WiFi.channel()
        + "\nRSSI of the network: " + WiFi.RSSI() + "dbm\n802.11 variant: " + variant);
}
else if(req == "STOP"){
    string ="STOP\n";
}
else{
    Serial.println(req+"|");
}
}
req= "";
}while(string != "STOP\n");

Serial.println("closing connection");
client.stop();

//Serial.println("wait 5 sec...");
delay(5000);
}
```