



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

TRABAJO DE FINAL DE GRADO

GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA

**INTEGRACIÓN DE PLATAFORMA ROBÓTICA MÓVIL EN SISTEMA DE
ENSAMBLAJE ACADÉMICO**



Anexos

Autor/a: Hidalgo Vélez, Justin Geovanny
Director/a: Ponsa Asensio, Pedro
Co-Director/a: Tornil Sin, Sebastián
Convocatoria: Junio 2021

Índice

ANEXO A	1
ANEXO B	2
ANEXO C	3

Anexo A

```
1 /*
2  * Programa para comunicación UDP entre una placa NodeMCU ESP32 y un PC usando un repetidor WiFi
3  * Justin Hidalgo Vélez - 30/05/2021
4  * Universidad Politécnica de Cataluña
5  */
6
7 /*Librerías*/
8 #include <WiFi.h>
9 #include <WiFiUdp.h>
10
11 /*Configuración WiFi*/
12 WiFiUDP Udp; //Asignación de un nombre a la comunicación UDP
13 char udp_buffer[255]; //Buffer para los paquetes entrantes hasta 255 bytes
14 unsigned int puerto_udp = 5219; //Puerto local UDP del NodeMCU
15 const char * ssid = "WiFi-Repeater"; //Nombre de la red wifi
16 const char * password = "123456789"; //Contraseña de la red wifi
17
18 /*Variables*/
19 int LED = 14; //Pin del LED
20 int acknowledged = 1; //Mensaje de confirmación para paquete recibido
21
22 void setup()
23 {
24   Serial.begin(115200);
25   pinMode(LED, OUTPUT);
26   Serial.println();
27   Serial.printf("Conectando a %s ", ssid);
28   WiFi.begin(ssid, password); //Inicializa la configuración de la biblioteca WiFi
29   while (WiFi.status() != WL_CONNECTED)
30   {
31     delay(500);
32     Serial.print(".");
33   }
34   Serial.println("Conectado correctamente!");
35   Udp.begin(puerto_udp); //Inicializa la biblioteca y la configuración de la red Wi-Fi UDP, escuchando en el puerto ()
36   Serial.println();
37   Serial.printf("Escuchando en la IP %s, puerto UDP %d", WiFi.localIP().toString().c_str(), puerto_udp);
38 }
39
40 void loop()
41 {
42   int paquete = Udp.parsePacket();
43
44   if (paquete)
45   {
46     // receive incoming UDP message
47     Serial.println();
48     Serial.printf("Received %d message from %s, port %d", packetSize, Udp.remoteIP().toString().c_str(), Udp.remotePort());
49
50     int len = Udp.read(udp_buffer, 255);
51
52     if (len > 0)
53     {
54       udp_buffer[len] = 0;
55     }
56     Serial.println();
57     Serial.printf("UDP message: %s", ReceivedMessage);
58
59     digitalWrite(LED, HIGH);
60     delay(250);
61     digitalWrite(LED, LOW);
62
63     udp.beginPacket(udp.remoteIP(), udp.remotePort());
64     udp.write(acknowledged);
65     udp.endPacket();
66   }
67 }
```

Anexo B

```

1 /*
2 Programa para comunicaci3n UDP entre una placa NodeMCU ESP32 y ROBOTINO mediante Acces Point TP-Link
3 Justin Hidalgo V3lez - 30/05/2021
4 Universidad Polit3cnica de Catalu1a
5 */
6
7 /*Librerias*/
8 #include <WiFi.h>
9 #include <WiFiUdp.h>
10
11 /*Configuraci3n WiFi*/
12 WiFiUDP udp; //Asignaci3n de un nombre a la comunicaci3n UDP
13 unsigned char udp_buffer[36]; //Buffer para los paquetes entrantes hasta 36 bytes de entrada
14 unsigned int puerto_udp = 2000; //Puerto local UDP del NodeMCU
15 const char * ssid = "Robotino.400.103.2.4GHz"; //Nombre de la red wifi
16 const char * password = "robotino"; //Contrase1a de la red wifi
17
18 IPAddress ip_local(192, 168, 0, 150); //IP local del NodeMCU
19 IPAddress gateway(192, 168, 0, 1);
20 IPAddress subnet(255, 255, 255, 0);
21
22 /*Variables*/
23 int LED = 14; //Pin del rele
24 int wifi = 33; //Pin de verificaci3n de conexi3n WiFi
25 int flag = 0; //Variable para detectar el cambio de 1 a 0 en el mensaje udp
26 int acknowledged = 1; //Mensaje de confirmaci3n para paquete recibido
27
28 void setup()
29 {
30 Serial.begin(115200);
31 pinMode(LED, OUTPUT);
32 pinMode(wifi, OUTPUT);
33 WiFi.config(ip_local, gateway, subnet);
34 delay(100);
35 conectarWifi();
36 udp.begin(puerto_udp); //Inicializa la biblioteca y la configuraci3n de la red Wi-Fi UDP, escuchando en el puerto ()
37 Serial.printf("Escuchando en la IP %s, puerto UDP %d", WiFi.localIP().toString().c_str(), puerto_udp);
38 }
39
40
41 void loop()
42 {
43 int paquete = udp.parsePacket(); //tama1o del paquete si no se envia nada es 0
44 if (paquete) //si se ha recibido algo
45 {
46 Serial.println();
47 Serial.printf("Mensaje de %d bytes recibido de %s, por el puerto %d", paquete, udp.remoteIP().toString().c_str(), udp.remotePort());
48
49 int len = udp.read(udp_buffer, 36); //lee 36 bytes y los almacena en el buffer
50 Serial.println();
51 Serial.printf("Mensaje UDP: %d", udp_buffer[4]); //solo nos interesa el byte 4
52
53 if ((udp_buffer[4] == 1) && (flag == 0)) //si el byte 4 es 1 y el flag es 0
54 {
55 digitalWrite(LED, HIGH);
56 flag = 1;
57 }
58 if ((udp_buffer[4] == 0) && (flag == 1)) //si el byte 4 es 0 y el flag es 1
59 {
60 digitalWrite(LED, LOW);
61 flag = 0;
62 }
63
64 udp.beginPacket(udp.remoteIP(), udp.remotePort());
65 udp.write(acknowledged);
66 udp.endPacket();
67 }
68 }
69
70 void conectarWifi()
71 {
72 WiFi.begin(ssid, password); //Inicializa la configuraci3n de la biblioteca WiFi
73 Serial.println("Conectando a la red: ");
74 Serial.print(ssid);
75
76 while (WiFi.status() != WL_CONNECTED)
77 {
78 Serial.print('.');
79 delay(200);
80 }
81 Serial.println("Conectado correctamente");
82 digitalWrite(wifi, HIGH);
83 }

```

Anexo C

