#include <ESP8266WiFi.h>

#include <WiFiUdp.h>

const char\* ssid = "WiFi\_Name"; // Wi-Fi access point

const char\* password = "WiFi\_Pwd"; // Wi-Fi Access point password

unsigned int localUdpPort = 80; // local port to listen on

char incomingPacket[1024]; // buffer for incoming packets

char buff[1024]; // temp buffer

int d0 = D0; int d1 = D1; // GPIO pins for read and write data

int d2 = D2; int d3 = D5;

volatile int wr\_1 = D6; volatile int rd\_1 = D7; // Write and Read enable pins

int clk = D4; // Input clock form FPGA

char wr\_reply[1024];

char rd\_reply[1024];

int byte\_1 = 'h';

int byte\_2 = '/';

int byte\_3 = '`';

char rd\_buff[255];

char nibb1[5] = "0010";

char nibb0[5] = "0011";

char tmp\_byte1[10];

WiFiUDP Udp;

void setup() {

pinMode(clk, INPUT);

pinMode(wr\_1, OUTPUT); pinMode(rd\_1, OUTPUT);

//pinMode(wr\_2, OUTPUT); pinMode(rd\_2, OUTPUT);

Serial.begin(115200);

Serial.println();

Serial.printf("Connecting to %s ", ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED)

{

delay(500);

Serial.print(".");

}

Serial.println(" connected");

Udp.begin(localUdpPort);

Serial.printf("Now listening at IP %s, UDP port %d\n", WiFi.localIP().toString().c\_str(), localUdpPort);

}

void loop() {

int packetSize = Udp.parsePacket();

if (packetSize)

{

// receive incoming UDP packets

Serial.printf("Received %d bytes from %s, port %d\n", packetSize, Udp.remoteIP().toString().c\_str(), Udp.remotePort());

int len = Udp.read(incomingPacket, 1024);

strcpy(buff, incomingPacket);

int w\_bit = buff[0] & (1 << 3); // To check whether Write bit is set

if (len > 0)

{

incomingPacket[len] = 0;

Serial.printf("UDP packet from client: %s\n", incomingPacket);

if (w\_bit) // if w\_bit is set, carry write operation

{

pinMode(d0, OUTPUT); pinMode(d1, OUTPUT); pinMode(d2, OUTPUT); pinMode(d3, OUTPUT); // Set the data pins as output

int bit\_0 = buff[1] & (1 << 0); // Check the data from incoming UDP packet

int bit\_1 = buff[1] & (1 << 1);

int bit\_2 = buff[1] & (1 << 2);

int bit\_3 = buff[1] & (1 << 3);

if (bit\_0) // Set the data out values to FPGA

{

digitalWrite(d0, HIGH);

}

else

{

digitalWrite(d0, LOW);

}

if (bit\_1)

{

digitalWrite(d1, HIGH);

}

else

{

digitalWrite(d1, LOW);

}

if (bit\_2)

{

digitalWrite(d2, HIGH);

}

else

{

digitalWrite(d2, LOW);

}

if (bit\_3)

{

digitalWrite(d3, HIGH);

}

else

{

digitalWrite(d3, LOW);

}

attachInterrupt(clk, Wr\_block1, RISING);

delay(100);

detachInterrupt(clk);

digitalWrite(wr\_1, HIGH);

delay(800);

digitalWrite(wr\_1, LOW); delay(500);

sprintf(wr\_reply, "%c" "%c" , byte\_1, byte\_2);

Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());

Udp.write(wr\_reply);

Udp.endPacket();

Serial.printf("UDP Packet from server: %s\n", wr\_reply);

delay(800);

byte\_1++;

}

else

{ // Read operation

pinMode(d0, INPUT); pinMode(d1, INPUT); pinMode(d2, INPUT); pinMode(d3, INPUT); // Set as input from FPGA (Read)

delay(500);

attachInterrupt(clk, Rd\_block2, FALLING);

delay(100);

detachInterrupt(clk);

digitalWrite(rd\_1, HIGH);

delay(600);

digitalWrite(rd\_1, LOW); delay(600);

int a = digitalRead(d0); int c = digitalRead(d2); // Read input from FPGA

int b = digitalRead(d1); int d = digitalRead(d3);

delay(800);

sprintf (rd\_buff, "%0i""%0i""%0i""%0i", d, c, b, a);

if (rd\_buff[0] == '0' && rd\_buff[1] == '0' && rd\_buff[2] == '0' && rd\_buff[3] == '0')

{

strcpy(tmp\_byte1, nibb0);

}

else

{

strcpy(tmp\_byte1, nibb1);

}

strcat(tmp\_byte1, rd\_buff);

char tmp\_byte2 = strtol(tmp\_byte1, NULL, 2);

sprintf(rd\_reply, "%c""%c", byte\_3 , tmp\_byte2);

Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());

Udp.write(rd\_reply);

Serial.printf("UDP Packet from server: %s\n", rd\_reply);

Udp.endPacket();

byte\_3++;

}

}

}

}

void Wr\_block1() {

Serial.println("Write operation");

}

void Rd\_block2() {

Serial.println("Read operation");

}