**Listing 1. Funkcja inicjująca komunikację oraz inicjowanie obsługi klientów**

void setup **(** void **)**

**{**

 // inicjalizacja pinow jako wyjscia

 pinMode **(** R1**,** OUTPUT **);**

 pinMode **(** R2**,** OUTPUT **);**

 pinMode **(** R3**,** OUTPUT **);**

 pinMode **(** R4**,** OUTPUT **);**

 // wyzerowanie pinow

 digitalWrite**(** R1**,** LOW **);**

 digitalWrite**(** R2**,** LOW **);**

 digitalWrite**(** R3**,** LOW **);**

 digitalWrite**(** R4**,** LOW **);**

 // stan domyslny przekaznikow z tablicy

 przekazniki\_zapal**();**

 // ustawienie uartu dla debugowania

 Serial**.**begin **(** 115200 **);**

 // polaczenie z siecia

 WiFi**.**begin **(** ssid**,** password **);**

 Serial**.**println **(** "" **);**

 // Oczekiwanie na polaczenie

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

 delay **(** 500 **);**

 Serial**.**print **(** "." **);**

 **}**

 // wyswietlenie informacji o polaczeniu

 Serial**.**println **(** "" **);**

 Serial**.**print **(** "Connected to " **);**

 Serial**.**println **(** ssid **);**

 Serial**.**print **(** "IP address: " **);**

 Serial**.**println **(** WiFi**.**localIP**()** **);**

 **if** **(** MDNS**.**begin **(** "esp8266" **)** **)**

 **{**

 Serial**.**println **(** "MDNS responder started" **);**

 **}**

 // przypisanie funkcji do danych jakie moze odebrac serwer

 server**.**on **(** "/"**,** handleRoot **);**

 server**.**on**(** "/relays"**,** przekazniki\_funkcja **);**

 server**.**on**(** "/rel"**,** json\_funkcja **);**

 server**.**onNotFound**(** handleNotFound **);**

 // uruchomienie serwera

 server**.**begin**();**

 // wyswietlenie informacji o rurchomieniu

 Serial**.**println **(** "HTTP server started" **);**

**}**

void loop **(** void **)**

**{**

 // obsluga klientow przez serwer

 server**.**handleClient**();**

**}**

**Listing 2. Sterowanie przekaźnikami oraz definicja strony w postaci stałej typu string**

const String strona **=**

"<html>\

 <head>\

 <title>ESP RELAY</title>\

 <style>\

 body { background-color: #cccccc; font-family: Arial, Helvetica, Sans-Serif; Color: #000088; }\

 </style>\

 </head>\

 <body>\

<p>Przekaznik nr 1:</p>\

<form action=\"relays\?rel1=on\" method=\"POST\">\

 <input type=\"submit\" value=\"Zapal\">\

</form>\

<form action=\"relays\?rel1=off\" method=\"POST\">\

 <input type=\"submit\" value=\"Zgas\">\

</form>\

<p>Przekaznik nr 2:</p>\

<form action=\"relays\?rel2=on\" method=\"POST\">\

 <input type=\"submit\" value=\"Zapal\">\

</form>\

<form action=\"relays\?rel2=off\" method=\"POST\">\

 <input type=\"submit\" value=\"Zgas\">\

</form>\

<p>Przekaznik nr 3:</p>\

<form action=\"relays\?rel3=on\" method=\"POST\">\

 <input type=\"submit\" value=\"Zapal\">\

</form>\

<form action=\"relays\?rel3=off\" method=\"POST\">\

 <input type=\"submit\" value=\"Zgas\">\

</form>\

<p>Przekaznik nr 4:</p>\

<form action=\"relays\?rel4=on\" method=\"POST\">\

 <input type=\"submit\" value=\"Zapal\">\

</form>\

<form action=\"relays\?rel4=off\" method=\"POST\">\

 <input type=\"submit\" value=\"Zgas\">\

</form>\

 </body>\

</html>"**;**

void handleRoot**()**

**{**

 server**.**send **(** 200**,** "text/html"**,** strona **);**

**}**

void handleNotFound**()**

**{**

 String message **=** "Site Not Found\n\n"**;**

 server**.**send **(** 404**,** "text/plain"**,** message **);**

**}**

void przekazniki\_funkcja**(** void **)**

**{**

 String message **=** "Otrzymalem:\n"**;**

 message **+=** "URI: "**;**

 message **+=** server**.**uri**();**

 message **+=** "\nMethod: "**;**

 message **+=** **(** server**.**method**()** **==** HTTP\_GET **)** **?** "GET" **:** "POST"**;**

 message **+=** "\nArguments: "**;**

 message **+=** server**.**args**();**

 message **+=** "\n"**;**

 // debug na konsoli

 **for** **(** uint8\_t i **=** 0**;** i **<** server**.**args**();** i**++** **)**

 **{**

 message **+=** " " **+** server**.**argName **(** i **)** **+** ": " **+** server**.**arg **(** i **)** **+** "\n"**;**

 **}**

 Serial**.**println **(** message **);**

 // rozkodowanie komendy

 **for** **(** uint8\_t i **=** 0**;** i **<** server**.**args**();** i**++** **)**

 **{**

 **if(** server**.**argName**(** i **)[** 0 **]** **==** 'r' **&&** server**.**argName**(** i **)[** 1 **]** **==** 'e' **&&** server**.**argName**(** i **)[** 2 **]** **==** 'l' **)**

 **{**

 // zabezpieczenie przed przekroczeniem indexu

 **if(** server**.**argName**(** i **)[** 3 **]** **-** '1' **<** 4 **)**

 **{**

 tablica\_relays**[** server**.**argName**(** i **)[** 3 **]** **-** '1' **]** **=** **(**server**.**arg**(** i **)** **==** "on" **?** '1' **:** '0'**);**

 **}**

 **}**

 **}**

 // wykonanie akcji

 przekazniki\_zapal**();**

 // przeslanie strony glownej

 handleRoot**();**

**}**