Mikrosterownik Wi-Fi

Prezentowane urządzenie to niewielki dwukanałowy sterownik działający jak miniserwer www. Wyróżnia się przede wszystkim łatwym sposobem konfiguracji – nie wymaga modyfikowania kodu programu dla ustawienia dostępu do routera, obsługa i konfiguracja odbywa się poprzez stronę internetową. Ponadto wyposażony jest w wejście binarne, które może posłużyć do monitorowania sterowanego obwodu.

Budowa

Na rysunku 1 pokazany jest schemat ideowy urządzenia. Wszystkie funkcje sterownika nadzoruje mikrokontroler modułu ESP01. Moduł montowany jest w ośmiostykowym złączu SV1, którego wyprowadzenia pełnią następujące funkcje:

- 2, 7 doprowadzenia napięcia zasilania o poziomie 3,3 V i prądzie 100 mA
- 5 wyjście GPIO2 wykorzystane do sterowania przekaźnikiem PK2
- 3 wyprowadzenie GPIO0 o dwu funkcjach: wykorzystane do sterowania przekaźnikiem PK1 lub do wprowadzania modułu ESP01 w tryb programowania

Dodatkowe materiały do pobrania ze strony www.media.avt.pl

W ofercie AVT* AVT-5<u>685</u>

Podstawowe parametry:

- wyjścia przekaźnikowe przełączne
- o obciążalności 10 A/250 VAC, wejście binarne, optoizolowane,
- aktywowane napieciem 3,3...5 V,
- zasilanie napięciem stałym 4,5…6 V
- o wydajności min. 250 mA, możliwość pracy w sieci Wi-Fi lub jako samodzielny punkt dostępowy AP,
- sterowanie z poziomu przeglądarki internetowej

FIUJEKLY	pokrewne na www.meuia.avc.pi.
AVT-5675	Moduł przekaźników sterowanych prze
	Wi-Fi (EP 4/2019)
AVT-5662	Kolorofon z Wi-Fi (EP 2/2019)
AVT-5530	Regulator natężenia oświetlenia
	- Wi Fi (FD 1/2016)

z Wi-Fi (EP 1/2016)
Lukngal Elektroniczne zestawy do sanddzielnego montażu.
Wymagana umiejętność lutowaniai
Podstawową wersją zestawu jest wersją [B] nazywana potocznie KIT-em (z ang. zestaw). Zestaw wersji [B] zautera elementy elektroniczne (w twp [UK] - jeśli występuje w projekcie), które należy samodzielnie vlutować w dołączon płytkę drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie ktu.
Mają na uwadze różne potrzeby naszych klientów, oferujemy dołatkowe wersje:
wersją [C] - zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wlutowane w płytkę PCB)
wersją [A] - płytka drukowana bez elementów i dokumentacji kliy w ktorych występuje układ scalony wymagający zaprogramowania, mają następujące dodatkowe wersje:
wersją [A] - płytka drukowana [A] + zaprogramowa układ [UK] i dokumentacja
wersją alkaj - zaprogramowany układ
Nie każdy zestaw Avt występuje we wszystkich wersjach Każda wersja napizh pdf Podczas składnia zamówienia upewnij się, którą wersję zamawiazsi
http://sklep.avt.pl, osoby zainteresowane zakupem płytek drukowanych (PCB) prosimy o kontakt via e-mail: kity@avt.pl.



- 1 wyprowadzenie URXD o dwu funkcjach: jako wejście binarne do badania stanu napięcia wejściowego lub jako wejście RXD do transmisji danych podczas programowania modułu
- 8 UTXD wyjście sygnału TXD do transmisji danych podczas normalnej pracy i w czasie programowania modułu
- 6 CH PD wejście wprowadzające moduł w tryb obniżonego poboru mocy, nieaktywne i na stałe podciągnięte poprzez opornik do poziomu wysokiego
- 4 RST wejście zerowania podciągnięte poprzez opornik do poziomu wysokiego z podłączonym przyciskiem RESET.

Na schemacie znajdują się dodatkowe szpilki JP1...JP4, które pozwalają na takie skonfigurowanie połączeń na płytce, że możliwe jest programowanie modułu ESP01 zamontowanego na płytce sterownika. Oto dokładny opis:

JP1 – zwora w pozycji 1-2: normalna praca; zwora w pozycji 2-3: tryb programowania modułu.

JP3 - zwora w pozycji 1-2: tryb programowania modułu; zwora w pozycji 2-3: normalna praca.

JP4 – złącze, które służy do podłączenia zewnętrznego konwertera sygnału USB-UART.

Szpilki JP2 nie są związane z programowaniem modułu, założenie zwory powoduje wymuszenie pracy mikrosterownika jako punktu dostępowego AP

Do złącza JP4 można dołączyć konwerter o poziomach 0...3,3 V (NIE 0...5 V!). W trybie programowania modułu złączem przesyłane są dane z zewnętrznego komputera do zapisu w pamięci FLASH kontrolera

modułu. Podczas normalnej pracy wyprowadzeniem JP4-2 (TXD out) transmitowane są komunikaty tekstowe o stanie sterownika, które można wyświetlić na ekranie komputera za pomoca dowolnego programu terminalowego.

Układy wykonawcze, sterujące dwoma przekaźnikami, są identyczne. Składają się ze sterującego przekaźnikiem tranzystora NPN, diody LED sygnalizującej świeceniem załączenie i diody tłumiącej przepięcia indukowane na cewce przekaźnika w czasie przełączania.

Transoptor OK1 zastosowano do separacji i zabezpieczenia wejścia binarnego, do którego doprowadzane jest badane napięcie. Opornik R7 ma tak dobraną wartość, żeby możliwa była detekcja napięcia wejściowego o poziomie 3,3...5 V. Ze względu na zastosowane przekaźniki o nominalnym napięciu pracy 5 VDC, zakres napięcia stałego użytego do zasilania mikrosterownika może się mieścić w przedziale od 4,5 do 6 V. Przy wyższym napięciu przekaźniki w czasie załączenia mogą się nadmiernie nagrzewać. Zasilanie dołączane jest do kostki zaciskowej X1. Dioda D1 zabezpiecza przed podaniem napięcia zasilającego o omyłkowej, odwrotnej polaryzacji.

Zaprogramowanie modułu ESP01

Program sterownika napisany został w środowisku Arduino z API przystosowanym do procesora ESP8266 (zamontowanego na płytce modułu ESP01). Zapis programu do procesora można przeprowadzić z poziomu Arduino po doinstalowaniu dodatkowych narzędzi. Są to:



Rysunek 1. Schemat sterownika

 dodatkowe biblioteki dla płytki ESP8266. Korzystając z menu Plik → Preferencje, w polu Dodatkowe adresy URL do menadżera płytek: trzeba wpisać http://bit.ly/2G476lA.

W polach zakładki *Narzędzia* trzeba wybrać:

- Płytka: "Generic ESP8266 Module"
- Flash Mode: "DIO"
- Flash Frequency "40MHz"
- CPU Frequency: "80MHz"
- Flash Size "512K (64K SPIFFS)"
- narzędzie do zapisu plików w obszarze SPIFFS. Należy pobrać plik instalacyjny spod adresu http://bit.ly/2Pg80KM. Należy rozpakować pobrany plik w katalogu Arduino IDE (C:\Program Files(x86)\ Arduino\tools). Po otwarciu Arduino IDE w zakładce Narzędzia powinna się pojawić dodatkowa pozycja ESP8266 Sketch Data Upload.

Po przygotowaniu narzędzi należy w Arduino IDE otworzyć projekt mikrosterownika z plikami źródłowymi:

Formatowanie_SHTML.h kontrola_SHTML.ino miniARS3_IO2.ino ObslugaEEPROM.h ObslugaEEPROM.ino ObslugaTagow binarnychIO.ino

Dla sprawdzenia kompletności plików projektu można przeprowadzić kompilację wybierając z menu Szkic → Weryfikuj/Kompiluj lub Ctr+R. Gdy kompilacja przebiegnie bez błędu można przystąpić do zapisu oprogramowania do pamięci FLASH. Do gniazda SV1 płytki mikrosterownika wkładamy moduł ESP01 tak aby ścieżka anteny na module była skierowana w tę samą stronę co jej symbol umieszczony na płytce sterownika. Do złącza JP4 podłączamy wyprowadzenia konwertera USB-UART np. AVT1775 tak, aby wyprowadzenie sygnału Tx konwertera łączyło się z wejściem JP4-3 RXD in, wyprowadzenie sygnału Rx konwertera z JP4-2 TXD out, należy także połączyć wyprowadzenia masy GND. Następnie podłączamy gniazdo USB konwertera do wolnego portu USB komputera a na płytce sterownika ustawiamy zwory: JP1 2-3, JP3 1-2. Do złącza X1 płytki sterownika podłączamy napięcie zasilania, obie diody LED1, LED2 powinny się zaświecić.

W menu Arduino IDE wybieramy Szkic → Wgraj lub Ctr+U. Rozpocznie się ponowna kompilacja i proces zapisu. Jedna z diod zamontowanych na module ESP01 powinna zacząć migotać. Jeżeli wszystko pójdzie dobrze, dioda na module gaśnie a w polu komunikatów Arduino IDE nie powinny pojawić się żadne ostrzeżenia. Program został zapisany do pamięci FLASH.

Ostatnim krokiem jest zapis danych do pamięci SPIFFS kontrolera ESP8266. Dane są treścią generowanych stron, które mają być wyświetlane przez przeglądarkę internetową po połączeniu z mikrosterownikiem. Pliki do zapisu należy umieścić w katalogu plików projektu ale w wydzielonym podkatalogu o nazwie *data*. W przypadku tego projektu zawartość podkatalogu jest następująca: /data

index.shtml
param_log_do_sieci.html
switch_off.png
switch_on.png
test_in.shtml

Jak widać są to źródła 3 podstron w standardowym HTML-u (o nietypowym rozszerzeniu .shtml za chwilę) oraz pliki dwóch obrazków .png które będą wyświetlane na jednej ze stron. Cała zawartość podkatalogu *data* ma trafić do wydzielonego obszaru SPIFFS w pamięci FLASH kontrolera ESP8266. Zapis danych do obszaru SPIFFS nie niszczy programu wpisanego do FLASH tak więc dane do wyświetlania stron generowanych przez moduł mogą być wielokrotnie modyfikowane i wgrywane bez konieczności powtórnego zapisu programu.

Ustawienia zwór przed zapisem do obszaru SPIFFS są takie same jak podczas zapisu programu do FLASH-a. Przed rozpoczęciem zapisu należy nacisnąć przycisk RESET na płytce mikrosterownika, obie diody LED powinny być zaświecone. Z menu Arduino IDE należy wybrać Narzędzia → ESP8266 Sketch Data Upload. Powinien zostać wyświetlony komunikat: Uploading 65536 bytes from to flash at 0x0006B000 i rozpoczyna się zapis do obszaru SPIFFS. Tak jak poprzednio nie powinny pojawić się na koniec żadne komunikaty ani ostrzeżenia. Należy odłączyć zasilanie i przestawić zwory w pozycje: JP3 2-3, JP1 1-2. Mikrosterownik jest gotowy do pracy.

Tryb AP i Stacji

Komunikacja z mikrosterownikiem poprzez sieć Wi-Fi może odbywać się w jednym z dwu trybów: jako Stacja lub jako samodzielny punkt dostępowy AP. W trybie Stacji sterownik automatycznie loguje się do wybranej sieci podając jej nazwę i hasło. Jeżeli są prawidłowe router nadzorujący sieć, w procesie DHCP, automatycznie przydzieli sterownikowi unikalny numer IP w obrębie sieci. Od tej chwili można się ze sterownikiem łączyć w standardowy

Mikrosterownik Wi-Fi

sposób. Po wpisaniu adresu IP w obszarze paska przeglądarki, nastąpi połączenie i wyświetlenie wygenerowanej przez sterownik strony. Tryb Stacji zostaje wywołany po Resecie lub włączeniu urządzenia jeśli zwora JP2 będzie rozwarta. Tryb Stacji jest wygodny dla użytkownika ponieważ umożliwia szybki dostęp do innych urządzeń i usług np. dostęp do internetu, w ramach tej samej sieci. Gdy jednak brak dostępu do sieci bo np. sterownik znajdzie się poza zasięgiem domowego routera lub nie jest znany przydzielony w obrębie sieci adres IP, należy skorzystać z trybu samodzielnego punktu dostępowego AP.

Tryb AP zostaje wywołany po Resecie lub włączeniu urządzenia jeśli zwora JP2 będzie zwarta. W tym trybie żeby połączyć się z mikrosterownikiem trzeba wykonać 2 kroki:

- Na komputerze lub telefonie komórkowym należy wyświetlić listę będących w zasięgu sieci Wi-Fi. W trybie AP mikrosterownik zgłasza się jako sieć o nazwie *ARS3_miniAP* z otwartym dostępem bez konieczności podawania hasła. Należy wybrać tą sieć.
- W pasku przeglądarki trzeba wpisać adres IP: 192.168.4.1 w trybie AP sterownik przydzieli sobie zawsze ten właśnie aders.
 Po wpisaniu adresu nastąpi połączenie

i wyświetlenie wygenerowanej przez sterownik strony.

Sterowanie pracą mikrosterownika

Użytkownik powinien mieć możliwość kontroli nad 3 stanami pracy sterownika:

- załączeniem i wyłączeniem przekaźników, których styki są wyprowadzone do złącz X2 WY1 i X3 WY2 oraz możliwością odczytu aktualnego stanu tych wyjść
- odczytem stanu wejścia binarnego
- ustawieniem nazwy i hasła dostępu do sieci gdy mikrosterownik pracuje w trybie Stacji.

Wszystko to dostępne z poziomu przeglądarki dla której treść stron HTML do wyświetlenia generuje sam mikrosterownik. Treść i forma stron mogą być zmieniane przez

Rysunek 3. Wygląd strony monitorującej stan wejścia

użytkownika. Teraz jednak opiszę jak działa sterownik ze stronami przygotowanymi dla tego projektu.

Na **rysunku 2** widać wygląd strony *index.* shtml, która wyświetlana jest jako pierwsza po połączeniu przeglądarki z mikrosterownikiem. Poniżej nazwy urządzenia wyświetlany jest jego bieżący adres IP oraz informacja, że połączenie odbywa się w trybie AP. W trybie Stacji adres IP będzie adresem przydzielonym w obrębie sieci.

Poniżej umieszczone są dwa duże przyciski sterujące stanem wyjść WY1 i WY2. Kolor biały przycisku oznacza, że wyjście jest w stanie wyłączenia. Kolor niebieski i czerwony oznaczają, że odpowiednie wyjście jest załączone. Stan wyjścia zmienia się z kolejnym kliknięciem przycisku. Duży przycisk ODŚWIEŻ wymusza odświeżenie wyświetlanej strony HTML. Na samym dole z lewej strony znajdują się linki do kolejnych podstron pozwalających na odczyt stanu wejścia oraz na zaprogramowanie parametrów dostępu do sieci w trybie Stacji.

Na **rysunku 3** pokazany został wygląd strony na której wyświetli się stan wejścia. Jeżeli na złącze X5-2 podane zostanie napięcie o poziomie równym lub większym od 3,3 V, zadziała tranzystor w transoptorze OK1 a na stronie zostanie wyświetlony symbol zamkniętego przełącznika. Jeśli napięcie nie jest podawane wyświetli się symbol otwartego przełącznika.

W trybie AP trzeba pamiętać aby po włączeniu mikrosterownika zdjąć zworę JP2,

Rysunek 2. Wygląd strony index.shtml

w przeciwnym wypadku będzie na stałe wyświetlany symbol przełącznika zamkniętego.

Na **rysunku 4** pokazano wygląd strony HTML wygenerowanej przez mikrosterownik, służącej do podania nazwy i hasła logowania do sieci w trybie pracy Stacja. Po naciśnięciu klawisza **PRZESŁANIE** dane logowania zostają przesłane z przeglądarki do sterownika i tam zapisane w jego pamięci nieulotnej.

Odczyt IP nadanego mikrosterownikowi w trybie pracy Stacja

Po pierwszym zalogowaniu się do nowej sieci, sterownikowi zostaje nadany nowy IP w ramach procesu DHCP. Użytkownik musi poznać ten numer, żeby w ogóle połączyć się ze sterownikiem pracującym w trybie Stacji. Jeżeli ma dostęp do routera kontrolującego lokalną sieć Wi-Fi, może przy pomocy narzędzi systemowych podejrzeć w routerze nowo dodany adres IP. Jeżeli istnieje możliwość podpięcia się do złącza szeregowego TXD out mikrosterownika, można je podłączyć do portu USB komputera i przy pomocy dowolnego programu terminalowego wyświetlić komunikaty które wysyła. Pracując w trybie Stacji, mikrosterownik po włączeniu zasilania lub Resecie prześle informacje o swoim statusie a także numer nadanego IP.

Istnieje także procedura zdalnego pobrania informacji o nadanym numerze IP. W tym celu należy:

- włączyć mikrosterownik w trybie AP,
- na komputerze lub telefonie wybrać sieć *ARS3_miniAP*, która jest nazwą punktu dostępowego mikrosterownika,
- w opisany wcześniej sposób przesłać do sterownika parametry logowania do wybranej sieci np. własnej sieci domowej,
- nie zmieniając ustawień sieci na komputerze lub telefonie, wyłączyć napięcie zasilające, zdjąć zworę JP2 sterownika i ponownie zasilić sterownik, który znajdzie się w trybie Stacji,
- na telefonie lub komputerze odświeżyć wyświetlaną stronę mikrosterownika. Na górze strony powinien pojawić się napis *Stacja IP:* i dalej nadany przez router numer w obrębie sieci,
- komputer lub telefon przełączyć na prace w obrębie sieci do której został dołączony mikrosterownik. Od tej chwili możliwa będzie komunikacja ze sterownikiem po wpisaniu odczytanego numeru IP.

Modyfikowanie generowanych przez sterownik stron HTML

Źródła generowanych do wyświetlenia na przeglądarce stron HTML, arkusze stylów, obrazki – wszystko to umieszczone jest jako pliki w obrębie pamięci SPIFFS kontrolera modułu ESP01. Bez ingerencji w program sterujący można je zmieniać personalizując wyświetlane strony mikrosterownika. Zmieniając treść plików, włączając w to dodawanie nowych podstron, należy pamiętać o dwu sprawach: całkowita objętość treści do wyświetlenia musi się zmieścić w 64 kilobajtach, taki jest rozmiar pamięci SPIFFS. A drugą sprawą w przypadku stron interaktywnych czyli takich, których wygląd będzie się zmieniał w zależności od stanu wejścia i wyjść mikrosterownika, jest przestrzeganie kilku zasad prostego API.

Jako strony do wyświetlenia najlepiej się sprawdza czysty HTML. Ze względu na ograniczenia przepustowości toru Wi-Fi modułu ESP01, kod powinien mieć jak najmniejszą objętość w przeciwnym wypadku strony będą się długo ładowały. Poza tym ograniczeniem nie ma przeszkód by stworzyć dowolną, własną stronę mikrosterownika zawierającą np. skrypty JavaScript.

Sterownik będzie przesyłał do wyświetlenia do przeglądarki, dwa rodzaje stron: statyczne zawsze wyglądające tak samo i dynamiczne, których wygląd będzie się zmieniał w zależności od stanu mikrosterownika. Te pierwsze mają standardowe rozszerzenie .html natomiast pliki stron dynamicznych mają rozszerzenie .shtml. Jako przykład strony dynamicznej niech posłuży plik strony *index.shtml*, która wyświetla się

Server HTML ESP8266 × +	
← → C O Niezabezpieczona 192.168.4.1/param_log_do_sieci.html	☆ ⊖ :
Parametry logowania do sieci	
SSID (nazwa sieci) 30 znaków max.	
hasło 30 znaków max.	
PRZESŁANIE	

<u>Powrót</u>

Rysunek 4. Wygląd strony służącej do ustawienia nazwy i hasła logowania do sieci

jako pierwsza po wpisaniu w pasku przeglądarki adresu sterownika. Strona o dokładnie takiej nazwie musi istnieć nawet jeżeli będzie jedyną stroną do wyświetlania. Strona jest dynamiczna ponieważ w dwu miejscach to co wyświetla zależy od aktualnego stanu mikrosterownika. Jest to określenie trybu pracy AP lub Stacji wraz z adresem IP, które będą się zmieniać zależnie od sytuacji. Drugim takim miejscem jest wygląd klawiszy sterujących wyjściami i zależny od ich aktualnego stanu. **Listing 1** zawiera cały kod strony *index.shtml.*

Oprogramowanie sterownika, przed wysłaniem do przeglądarki treści strony, sprawdza rozszerzenie nazwy pliku. Rozszerzenie .shtml oznacza, że zawartość pliku przed wysłaniem ma być zmodyfikowana, pliki o innym rozszerzeniu wysyłane są w takiej formie w jakiej zapisano je w pamięci SPIFS. Oprogramowanie mikrosterownika szuka w pliku .shtml miejsca wystąpienia takiej -->. W normalnym konstrukcji: <!-pliku .html jest to oznaczenie komentarza nie wpływającego na wyświetlaną treść, w pliku .shtml oznacza miejsce potencjalnych zmian w treści strony. Jeżeli bezpośrednio za znakami otwierającymi komentarz znajdzie się obsługiwany tag (w tym przypadku #tag_w_ txt1_32z), oprogramowanie sterownika wykona następujące działania:

- usunie w tekście modyfikowanej strony wszystkie znaki komentarza zastępując je spacjami,
- w to miejsce zostanie wpisany tekst będący opisem statusu pracy sterownika i odczytany numer IP.

Poniżej treść fragmentu strony przed ingerencją oprogramowania sterownika i po:

<!--#tag_w_txt1_32z

01234567890123456789012345678901--> Stacja IP:192.168.0.185

Cyfry za tagiem *0123...* mają charakter pomocniczy do wyznaczenia przestrzeni 32 znaków. Podobny mechanizm zastosowano do dynamicznej zmiany wyglądu klawiszy sterujących stanem wyjść WY1 i WY2. Klawisze są przyciskami typu *submit* dwu formularzy. Sytuacja jest trochę bardziej skomplikowana niż poprzednio ponieważ wygląd przycisków ma sygnalizować stan wyjść, a ich naciśnięcie w przeglądarce inicjuje zmianę stanu wybranego wyjścia na przeciwny. Z tego powodu w formularzu oprócz elementu typu *submit* wykorzystano także pole ukryte czyli element typu *hidden*. W tym przypadku algorytm działania oprogramowania mikrosterownika przed wysłaniem strony do przeglądarki jest następujący:

 za znakami otwierającymi komentarz wyszukiwany jest tag #tag_w_out1 oznaczający, że odnosi się do stanu WY1. Następnie badana jest cyfra za znakiem '=' określająca stan WY1, 0 wyłączone 1 włączone. Jeżeli rzeczywisty stan WY1 jest zgodny z cyfrą za znakiem '=' usuwane są znaki komentarza, w przeciwnym wypadku nic się nie dzieje. W ten sposób kolor wyświetlanego w przeglądarce klawisza odpowiada aktualnemu stanowi wyjścia. Jeżeli w przeglądarce klawisz zostanie naciśnięty, wysyła ona komunikat do mikrosterownika z nowym stanem klawisza: form name out1=0 co oznacza, że WY1 ma być wyłączone lub form_name_out1=1 co oznacza, że WY1 ma być załączone. Taki sam mechanizm działa w przypadku klawisza sterującego WY2. Na podobnej zasadzie działa strona dynamiczna test_in.shtml służąca do wyświetlania stanu wejścia binarnego. Jej źródło zostało zamieszczone na listingu 2.

Ponieważ stan wejścia jest tylko odczytywany nie trzeba tej informacji wysyłać do przeglądarki jako formularza. Może to być zwykły tekst lub jak to widać na Listingu2 link do wyświetlanego odpowiedniego obrazka. Podobnie jak w przypadku stanu wyjść, badana jest zgodność cyfry za tagiem z aktualnym stanem WE1. Jeżeli #tag_w_in1=0 i stan napięcia na WE1 jest wysoki, zostaną usunięte znaki komentarza a przeglądarka

```
Listing 1. Kod strony index.shtml
<!DOCTYPE html>
<html>
<head>
                           <title>Serwer HTML ESP8266</title>
<meta name="description" content="Mikro serwer ESP8266">
                           <meta http-equiv="content-type" content="text/html; charset=UTF-8">
<meta http-equiv="content-type" content="application/xhtml+xml; charset=UTF-8">
</meta http-equiv="content="application/xhtml+xml; charset=UTF-8">
</meta http-equiv="content="application/xhtml+xml; charset=UTF-8">
</meta http-equiv="content="application/xhtml+xml; charset=UTF-8">

                           <style>
                                                       input {
                                                                                  width: 90%:
                                                                                  padding: 16px 32px;
font-size: 60px;
                                                       p {font-size: 30px;}
                           </style:
</head>
<body style="background-color:#EAEAEA:"
                            e= background-color:#EAEAEA; >
<div style="text-align: center"><h1>Mikro serwer ESP8266</h1>
<b><!--#tag_w_txt1_32z 01234567890123456789012345678901--></b>
                                                       <b>sterowanie 2 wyjściami</b>
                                                       <div align=center:
                                                                                  <tr align=center

<form method="get" action="index.shtml">
<form method="get" action="index.shtml">
<!--#tag_w_out1=1 <input type="submit" value="wy1" style="background-color:#0000ff;color:#f2f2f2">
<input type="hidden" name="form_name_out1" value="0"/>-->
<!--#tag_w_out1=0 <input type="submit" value="w1" style="background-color:white;color:black">
<input type="hidden" name="form_name_out1" value=""u1"/>-->

                                                                                                              </form>
                                                                                                              <td width=50%
                                                                                                              <form method="get" action="index.shtml">
                                                                                                             <!--#tag_w_out2=1 <input type="submit" value="Wy2" style="background-color:#ff0000;color:#f2f2f2">
<input type="hidden" name="form_name_out2" value="0"/>-->
<!--#tag_w_out2=0 <input type="submit" value="Wy2" style="background-color:white;color:black">
<input type="hidden" name_out2" value="0"/>-->
                                                                                                              </form>
                                                                                                              </div>
                                                      <form method="get" action="index.shtml">
<br><br><br></pr>
                                                                                   <input type="submit" name="odczyt" value="ODŚWIEŻ"</pre>
                                                       style="backgroundcolor:Lavender;color:DarkCyan"
</form>
                                                       <br />
                           </div>
                           <hr />
                            <a href="test_in.shtml">0dczyt stanu wejść</a>
<a href="param_log_do_sieci.html">Parametry logowania do sieci</a>
</body>
</html>
```

wyświetli obrazek *switch_on.png.* W przeciwnym wypadku będzie wyświetlony obrazek *switch_off.png.* Uwaga wartość tagu odnosi się do rzeczywistego poziomu napięcia na wyprowadzeniu URXD pełniącego funkcję

Rysunek 5. Schemat płytki PCB

wejścia binarnego. Transoptor OK1 odwraca fazę i gdy napięcie na WE1 ma poziom wysoki na URXD podawany jest stan niski.

Z kolei na **listingu 3** pokazane zostało źródło zwykłej strony *param_log_do_sieci.html* służącej do podawania nazwy sieci i hasła potrzebnych do logowania gdy sterownik pracuje w trybie Stacji.

Do przesłania wprowadzonej w przeglądarce nazwy sieci i hasła wykorzystano formularz, którego nazwy pól wejściowych muszą mieć nazwę form_name_siec_nazwa i form_name_ siec_haslo. Ostatecznie przeglądarka wyśle do mikrosterownika transmisję w której znajdą się podane twoja nazwa się sieci i twoje hasło sieci:

form_name_siec_nazwa=twoja_
nazwa_sieci&form_name_siec_
haslo=twoje_haslo_sieci

Po odebraniu transmisji oprogramowanie mikrosterownika rozpozna zastrzeżone nazwy form_name_siec_nazwa i form_name_ siec_haslo dzięki którym zidentyfikuje przesłaną nazwę sieci i hasło po czym zapamięta je w części pamięci pełniącej funkcję pamięci EEPROM.

Tagi i nazwy zastrzeżone formularzy

Poniżej podaję wszystkie tagi i zastrzeżone nazwy formularzy na które reaguje oprogramowanie sterownika:

 form_name_out1 przesyłanie żądania ustawienia poziomu wyjścia WY1,

Wykaz elementów: Rezystory: R1, R3, R5, R8...R10: 10 kΩ R2, R4: 1 kΩ R6: 3 kΩ R7: 510 Ω Kondensatory: C1, C2, C4: 100 nF C3: 220 µF/16 V Półprzewodniki: D1: 1N5817 D2, D3: 1N4148 LED1, LED2: dioda LED 3 mm T1, T2: BC548 0K1: PC817 US1: LD1117-3.3V SV1: ESP01 Pozostałe: PK1, PK2: SRD-05VDC-SL-C S1: mikroswitch X1: DG301-5.0/2 X2, X3: DG301-5.0/3 X5: DG381-3.5/3 JP2: goldpin 1×2 + jumper JP1, JP3, JP4: goldpin 1×3 + jumper

PROJEKTY

```
Listing 2. Kod strony wyświetlającej stan wejścia
 DOCTYPE html>
<html>
            <head>
                       <title>Serwer HTML ESP8266 test wejść</title> <meta charset="utf-8" />
                        <meta http-equiv="Refresh" content="5">
                       <style>
                                   input {
width: 90%;
                                   padding: 16px 32px;
font-size: 60px;
                                      {font-size: 30px;}
                                    table {font-size: 30px;}
                       </style>
            </head>
            <body style="background-color:#EAEAEA;">
                       .e="background-color:#EAEAEA;">
<div style="text-align: center"><hl>Mikro serwer ESP8266</hl>
cb>odczyt stanu wejść (automatycznie i na żądanie)</b>
                                   <div align=center>
                                                WEJŚCIE 1
                                                           <!--#tag_w_in1=0 <img src="switch_on.png" width="64" height="64" alt="">-->
<!--#tag_w_in1=1 <img src="switch_off.png" width="64" height="64" alt="">-->
                                                           </div>
                                   </form>
                       </div>
                       <b><a href="index.shtml">POWRÓT</a></b>
            </hody>
</html>
```

parametr value=0 wyłącz, value=1 załącz

 form_name_out2 przesyłanie żądania ustawienia poziomu wyjścia WY2 parametr value=0 wyłącz, value=1 załącz,

- form_name_siec_ nazwa przesyłanie do zapamiętania SSID nazwy sieci w trybie Stacja,
- form_name_siec_ haslo przesyłanie do zapamiętania hasła sieci w trybie Stacja,
- form_name_siec_param sygnalizacja przesyłania nazwy i hasła sieci, niewykorzystywane,

.....

- tagi identyfikatory wyboru:
- tag_w_out1 identyfikator stanu wyjścia WY1 na stronie dynamicznej .shtml parametr =0 wyłączone lub =1 włączone
- tag_w_out2 identyfikator stanu wyjścia WY2 na stronie dynamicznej .shtml

- REKLAMA ·

Listing 3. Źródło strony służącej do podawania nazwy sieci i hasła CTYPE html> <html> <head> <meta charset="utf-8" / <title>Serwer HTML ESP8266</title> <style> input {
font-size: 25px; p {font-size: 30px;} </style> </head> <body> <input type="submit" name="form_name_siec_param" value="PRZESŁANIE" /> </form> </div> Powrót </body> </html>

parametr =0 wyłączone lub =1 włączone

- tag_w_in1 identyfikator stanu wejścia IN1 na stronie dynamicznej .shtml parametr =0 stan niski lub =1 stan wysoki

UWAGA! w mikrosterowniku zanegowane przez transoptor.

 tag_w_txt1_32z identyfikator wyświetlenia 32 znaków tekstu wyświetlanie tekstu określającego tryb pracy i aktualny adres IP.

Montaż i uruchomienie

Schemat płytki PCB wraz z opisem rozmieszczenia elementów pokazuje **rysunek 5.** Urządzenie zbudowane jest z kilkunastu elementów montowanych w sposób przewlekany więc montaż nie sprawi problemu nawet osobom mniej doświadczonym. Urządzenie jest gotowe do pracy po zmontowaniu i zaprogramowaniu modułu ESP01.

Ryszard Szymaniak

0 projektach, mini i soft dyskutuj na https://forum.ep.com.pl