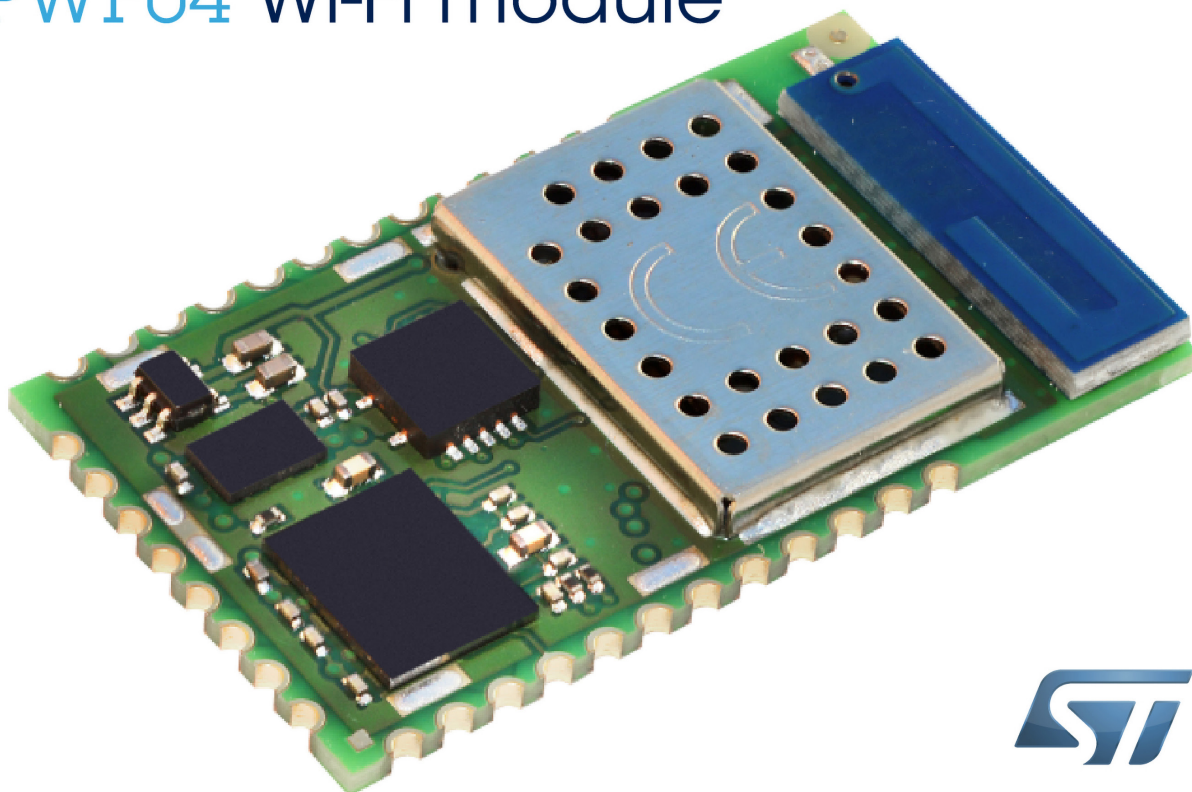


SPWF04 Wi-Fi module



Wi-Fi na STM32 z użyciem STM32CubeMX

Moduły Wi-Fi, podobnie jak inne moduły realizujące połączenia bezprzewodowe (Bluetooth, Thread itp.) są dość popularne i oferuje je wielu producentów. Jednym z nich jest SPWF01SA produkowany przez firmę STMicroelectronics.

Moduły Wi-Fi są elementami łączącymi zasoby sprzętowe z możliwościami programowymi niezbędnymi do pracy sterownika mikroprocesorowego w sieciach WLAN – Wireless Local Area Network. Przez zasoby sprzętowe rozumiemy kompletny transceiver pracujący w paśmie częstotliwości 2,4 GHz, lub/i 5 GHz oraz wbudowany wydajny mikrokontroler wspierający obsługę stosu TCP/IP i szyfrowanie transmisji WEP/WPA/WPA2.

Moduł jest umieszczony na płytce drukowanej przeznaczonej do montażu powierzchniowego. To rozwiązanie dobrze się sprawdza w nowo projektowanych urządzeniach. Jednak żeby można było przeprowadzić testy, chociażby w celu potwierdzenia przydatności w nowym projekcie ST przygotował moduł ewaluacyjny X-NUCLEO-IDW01M1 kompatybilny sprzętowo z modułami mikrokontrolerów serii STM32Nucleo.

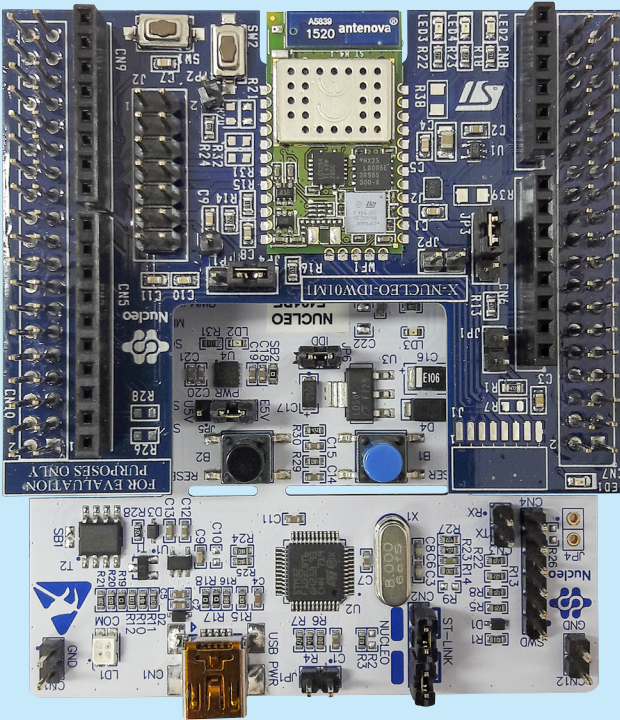
Moduły ewaluacyjne NUCLEO (moduły mikrokontrolerów i płytek rozszerzeń) są sprzętowo zgodne ze standardem Arduino UNO R3, ale żeby wykorzystać wszystkie możliwości sprzętowe mikrokontrolerów płytki NUCLEO wyposażono w dodatkowe złącza nazwane ST Morpho. Zestaw X-NUCLEO-IDW01M1 można łączyć z modułem mikrokontrolera

tylko za pomocą złącza Morpho. Na **fotografii 1** pokazano połączone moduły Wi-Fi X-NUCLEO-IDW01M1 i NUCLEO-F401RE.

Dla wszystkich zaawansowanych modułów X-NUCLEO ST przygotowuje i udostępnia programy demonstracyjne z kompletnymi projektami zawierającymi pliki źródłowe w C, pod różne środowiska projektowe. Tak też jest w przypadku modułu X-NUCLEO-IDW01M1. Dostępne są też gotowe pliki wynikowe do wgrania bezpośrednio do pamięci mikrokontrolera. Na **rysunku 2** przedstawiono blokowo strukturę budowy programów demonstracyjnego.

Komponenty programowe zawierają kompletne middleware do budowania aplikacji używającej SPWF01SA. Warstwa API pozwala na prostą konfigurację i pracę modułu. Dzięki możliwości używania programu STM32Cube można w prosty sposób dostosowywać aplikację do różnych typów mikrokontrolerów STM32. Dostarczone przez STMicroelectronics przykłady umożliwiają pracę modułu Wi-Fi z modułami NUCLEO-F103RB, F401RE i L053R8.

Jak już wspominałem, moduł SPWF01SA ma wbudowany mikrokontroler z rodziny STM32 Cortex-M3 z zapisanym w pamięci Flash firmware. To firmowe oprogramowanie jest rozwijane i jest możliwość jego



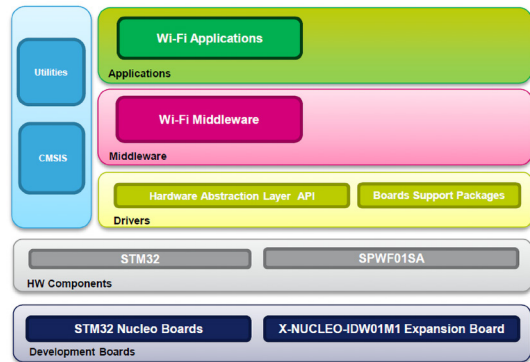
Fotografia 1. Połączone moduły X-NUCLEO-IDW01M1 i NUCLEO-F401RE

aktualizacji przez interfejs UART. Trzeba do pamięci mikrokontrolera modułu STM32 X-NUCLEO połączonego z X-NUCLEO-IDW01M1 wgrać skompilowany plik z katalogu FW_Upgrade_UART zawartego w paczce programowej z programami testowymi. Moduł mikrokontrolera STM32 Nucleo może się teraz łączyć z komputerem poprzez wirtualny UART via USB. Do komunikacji można użyć dowolnego programu terminalowego np. PuTTY, TeraTerm itp. Parametry transmisji: 115200 b/s, 8 b, none; 1 i Flow Ctrl: None. Po wysłaniu komendy AT+S.STS moduł odpowie informacją o wersji firmware wgranej do mikrokontrolera modułu SPWF01SA. Na jej podstawie można podjąć decyzję o aktualizacji.

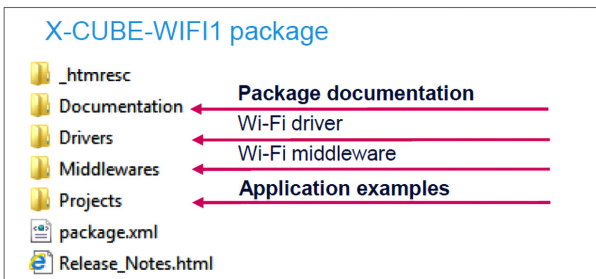
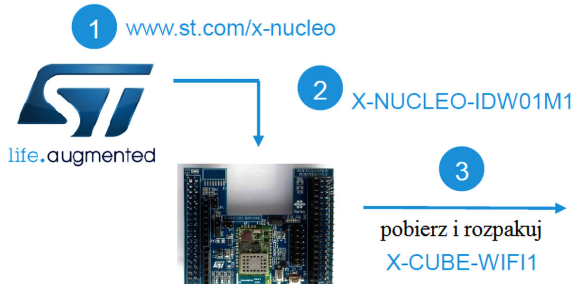
Najważniejsze właściwości SPWF01SA:

- Transceiver pracujący w paśmie 2,4 GHz zgodnie z IEEE 802.11 b/g/n
- Moduł radiowy spełnia wszystkie europejskie normy i dyrektywy w zakresie kompatybilności elektromagnetycznej
- Zintegrowana antena w wersji SPWF01SA.xy, lub zintegrowane złącze antenowe w wersji SPWF01SC.xy
- Moc TX 18,3 DSSS DSM 1 dBm, 13,7 dBm przy 54 MB/s OFDM
- Czulość RX -66 dBm przy 1 Mbps OFDM, -74,5 dBm przy 54 MB/S OFDM
- Interfejs UART (Host)
- Wbudowany mikrokontroler STM32 Cortex-M3 64 kB RAM i 512 kB Flash
- Zintegrowany stos TCP/IP: 8 jednoczesnych klientów TCP, lub UDP i 1 socket server
- 1 socket klient TLS/SSL wspierający protokół TLS1.2 z algorytmami szyfrowania AES, hash i algorytmy klucza publicznego (RSA, ECC)
- Serwer internetowy obsługujący dynamicznie strony internetowe
- Szyfrowanie transmisji WEP/WPA/WPA2
- Możliwość aktualizacji firmware za pośrednictwem UART

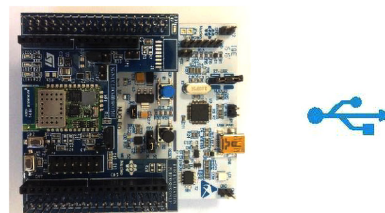
Overall Software Architecture



Rysunek 2. Struktura programu demonstracyjnego



4 pobierz i zainstaluj STM32 Nucleo ST-LINK/V2-1 USB driver



Rysunek 3. Pobieranie programów przykładowych

Procedurę aktualizowania firmware dokładnie opisuje dokument *X-NUCLEO-IDW01M1-FW upgrading over UART.pdf* dostępny z programami demonstracyjnymi w katalogu Documentation.

Zestaw programów demonstracyjnych oraz prostych programów pomocniczych pozwala użytkownikowi na konfigurowanie modułu SPWF01SA:

- do połączenia z punktem dostępowym Acces Point (AP),
- do pracy w trybach STA (Station), MiniAP (mini Acces Point), IBBS (pracy w konfiguracji ADHoc),
- do skanowania dostępnej sieci, wybór punktu dostępowego AP i połączenia z nim,

```
# Status & Statistics:
# version = 160129-c5bf5ce-SPWF01S
```

Rysunek 4. Przykładowa odpowiedź na komendę AT+S.STS

```

/*****
 *
 * X-CUBE-WIFI1 Expansion Software v3.1.1
 * X-NUCLEO-I040xx1 Wi-Fi Configuration.
 * Client-Socket Example
 *
 *****/

Do you want to setup SSID?(y/n):y

Enter the SSID:NETIA_MIK

Enter the password:Dd2c1814ad029

Enter the encryption mode(0:Open, 1:WEP, 2:WPA2/WPA2-Personal):2

Enter the Hostname (Apache Server): 192.168.1.4

```

Rysunek 5. Wprowadzanie przez konsolę danych do połączenia z siecią WLAN

```

/*****
 * Configuration Complete
 * Port Number:32000, Protocol: TCP/IP
 * Please make sure a server is listening at given hostname
 *****/

Initializing the wifi module NOW ...

Build Configuration:
Nucleo: NUCLEO-F401RE
X-Nucleo: X-NUCLEO-I0401M1
Interface: UART

WiFi Initialised and Ready..

End of Initialization..

```

Rysunek 6. Inicjalizacja modułu Wi-Fi

```

>>Connecting...
multimedia_domBSM16
www.uniedziwiedzia.net
NETIA_MIK

>>network present...connecting to AP...
*****
>>connected...

>>IP address is 192.168.1.3

>>model no is SPWF01SA1

```

Rysunek 7. Łączenie z siecią o SSID NETIA_MIK

Device Info -- DHCP Leases

	Hostname	MAC Address	IP Address	Expires In
moduł Wi-Fi	iwm-B8-D7-75	00:80:E1:B8:D7:75	192.168.1.3	23 hours, 56 minutes, 10 seconds
komputer	jablonskit-lap	AC:81:12:08:DF:2F	192.168.1.4	23 hours, 19 minutes, 27 seconds

Rysunek 8. Urządzenia pracujące w lokalnej sieci LAN

- do użycia połączenia TCP/UDP do otwierania, zamykania, oraz zapisywania/odczytywania socketów.

Przykład Client Socket

W tej aplikacji moduł Wi-Fi jest skonfigurowany jako STA (station) i musi się połączyć z punktem dostępowym AP. Tym punktem w czasie testu był router wykorzystywany łącznie się z Internetem za pomocą linii telefonicznej z wbudowanym interfejsem Wi-Fi. Połączenie wymaga podania SSID sieci, rodzaju szyfrowania transmisji radiowej i hasła dostępu. Po wgraniu programu do pamięci mikrokontrolera i połączeniu z portem USB komputera mamy możliwość używania konsoli znakowej przez wirtualny interfejs UART pracujący z prędkością transmisji 115200 b/s i parametrami transmisji Data: 8bit, Parity: None; Stop Bit: 1bit, Flow Ctrl: None. Ja użyłem programu TeraTerm. Na **rysunku 5** pokazano zrzut z ekranu w trakcie wpisywania przez konsolę danych potrzebnych do połączenia z siecią WLAN routera.

Po potwierdzeniu, że będziemy wprowadzać dane do konfigurowania transmisji wprowadzamy kolejno:

- identyfikator sieci SSID
- hasło dostępu
- rodzaj szyfrowania transmisji
- nazwę (adres ip) komputera pełniącego rolę serwera przy nawiązaniu połączenia klient-serwer

Po wprowadzeniu tych danych rozpoczyna się inicjalizowanie modułu Wi-Fi i jeżeli wszystko jest poprawne wyświetli się komunikat „WiFi Initialised and Ready, End of Initialization” – **rysunek 6**.

Po prawidłowej inicjalizacji wykonywane jest skanowanie sieci, wykrywanie jej identyfikatorów i próba łączenia z siecią o SSID wprowadzonym z konsoli – **rysunek 7**.

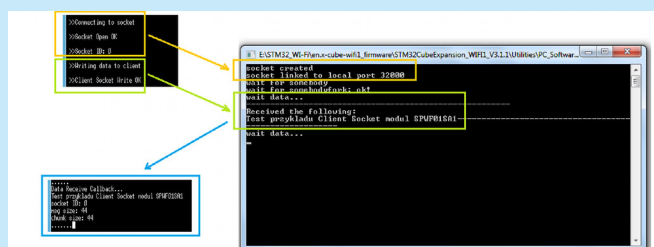
Kiedy połączenie z siecią WLAN jest zestawione, router przydziela modułowi adres IP – w naszym przypadku jest to adres 192.168.1.3. Na **rysunku 8** pokazano fragment zakładki DHCP strony konfiguracyjno-administracyjnej routera z tabelą zawierającą nazwy hostów pracujących w lokalnej sieci, ich adresy MAC i adresy IP. Nasz moduł zgłosił się jako *iwm-B8-D7-75* i jak już wspomniałem przydzielono mu adres 192.168.1.3.

```

>>Connecting to socket
>>Socket Open OK
>>Socket ID: 0
>>Writing data to client
>>Client Socket Write OK

```

Rysunek 9. Łączenie klient-serwer



Rysunek 10. Przesyłanie danych w przykładzie Client Socket

Listing 1. Zapisanie danych klient-serwer

```

case wifi_state_socket_write:
    HAL_Delay(500);

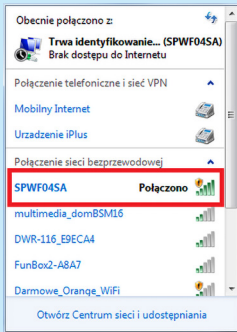
    printf(„\r\n >>Writing data to client\r\n");

    memset(echo, 0x00, 512);
    sprintf(echo, 50, „Test przykladu Client Socket modul SPWF01SA1");
    len = strlen(echo);
    /* Read Write Socket data */
    status = wifi_socket_client_write(socket_id, len, echo);

    if(status == WiFi_MODULE_SUCCESS)
    {
        printf(„\r\n >>Client Socket Write OK \r\n");
    }
    wifi_state = wifi_state_idle;

    break;

```



Rysunek 11. Wykrycie sieci modułu pracującego w trybie miniAP

```
Listing 2. Ustawienia miniAP
char * ssid = „SPWF04SA”;
char * seckey = „123456789”;
uint8_t channel_num = 6;
WiFi_Priv_Mode mode = WPA_Personal;
```

= 192.168.1.4 pracującym w tej samej sieci co moduł Wi-Fi. Ten program otwiera swój socket na porcie 32000 i czeka na połączenie klienta i po jego zrealizowaniu połączenia czeka na dane. Informacja o zrealizowaniu połączenia jest sygnałem dla klienta o konieczności wysłania danych do aplikacji server. Klient testowo wysłał do serwera łańcuch znakowy „Test przykładu Client Socket moduł SPWF01SA1”. Serwer po odebraniu danych odsyła je kontrolnie do klienta (echo). Na **rysunku 10** pokazano kolejne etapy zestawiania połączenia i przesyłania danych.

Na **listingu 1** zamieszczono fragment pętli głównej programu odpowiedzialny za przesłanie danych klient-serwer.

Przykład serwer tryb MiniAP

W tym przypadku moduł Wi-Fi jest skonfigurowany jako punkt dostępu Acces Point MiniAP. Podobnie jak w przypadku poprzedniego programu testowego jest możliwość łączenia z programem terminala znakowego się przez wirtualny port szeregowy UART o dokładnie takich samych parametrach transmisji. MiniAp ma domyślny SSID SPWF04SA, ustawione szyfrowanie WPA2 i hasło „123456789” – **listing 2**.

Dane konfiguracyjne można zmienić po restarcie mikrokontrolera, podobnie jak w przypadku poprzedniego programu przykładowego. Na **rysunku 11** pokazano okno z wykrytą siecią o SSID SPWF04SA.

Zrzut z ekranu terminala znakowego w trakcie inicjalizacji programu Server-Socket z domyślnymi ustawieniami został pokazany na **rysunku 12**.

W terminalu jest wypisywany adres IP i MAC adres. Moduł ma w sieci lokalnej na sztywno ustalony adres IP 172.24.117.1. Do testowania połączenia server-client użyjemy programu SocketTest V3.0.0 dostarczanego w paczce programów testowych w katalogu Utilites. Uruchamiamy go na komputerze połączonym przez Wi-Fi do sieci miniAP o SSID SPWF04SA. Najpierw trzeba otworzyć zakładkę Client i wpisać w pole IP Address adres IP miniAp równy 172.24.117.1.

W polu Port wpisujemy wartość 32000. Po wpisaniu tych danych klikamy na przycisk Connect i czekamy na połączenie z serwerem. Rolę serwera pełni moduł a rolę klienta komputer z uruchomionym programem SocketTest V3.0.0. Po połączeniu w polu Connected To pojawia się adres IP 172.24.117.1. – **rysunek 13**.

Po ustaleniu połączenia w konie Message można wpisywać tekst przesyłany do serwera. Naciśnięcie przycisku Send powoduje, że łańcuch znakowy jest przesyłany do serwera i można to zobaczyć w terminalu znakowym – **rysunek 14**.

Komputer (laptop) łączący się przez Wi-Fi i pracujący w tej sieci na przydzielony adres 192.168.1.4. Ten sam adres wprowadziliśmy jako Hostname (APACHE Server) w trakcie wprowadzania danych konfiguracyjnych. Pokazane to zostało na rysunku 5. Po nawiązaniu połączenia z AP i przydzieleniu adresu IP w sieci moduł Wi-Fi próbuje otworzyć połączenie (socket connection) ze zdalnym socketem.

Jeżeli próba połączenia przebiegnie poprawnie, to moduł otwiera własny zdalny socket. Do testowania połączenia klient-serwer można użyć testowego programu *server.exe* dostępnego w katalogu Utilites\tcp socket w paczce programów testowych. *Server.exe* musi być uruchomiony na komputerze PC o adresie Hostname

= 192.168.1.4 pracującym w tej samej sieci co moduł Wi-Fi. Ten program otwiera swój socket na porcie 32000 i czeka na połączenie klienta i po jego zrealizowaniu połączenia czeka na dane. Informacja o zrealizowaniu połączenia jest sygnałem dla klienta o konieczności wysłania danych do aplikacji server. Klient testowo wysłał do serwera łańcuch znakowy „Test przykładu Client Socket moduł SPWF01SA1”. Serwer po odebraniu danych odsyła je kontrolnie do klienta (echo). Na **rysunku 10** pokazano kolejne etapy zestawiania połączenia i przesyłania danych.

Na **listingu 1** zamieszczono fragment pętli głównej programu odpowiedzialny za przesłanie danych klient-serwer.

```
*****
* X-CUBE-WIFI1 Expansion Software V3.1.1
* X-NUCLEO-I040xx1 Hi-Fi Mini-AP Configuration.
* Server-Socket Example
*
*****
Do you want to setup SSID?(y/n):n

Module will connect with default settings.
*****
* Configuration Complete
* Please make sure a server is listening at given hostname
*****
Initializing the wifi module...

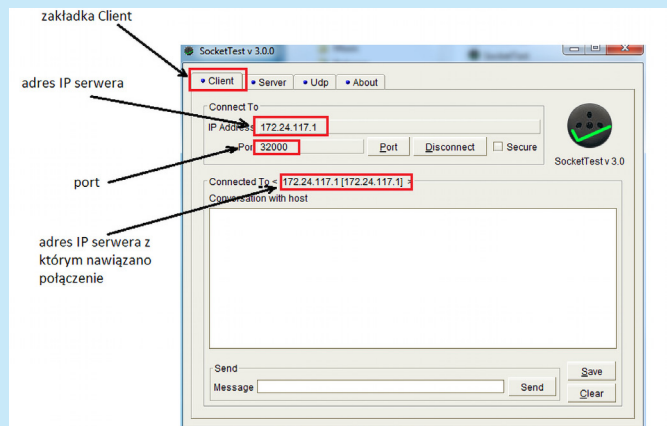
Build Configuration:
Nucleo: NUCLEO-F401RE
X-Nucleo: X-NUCLEO-I0401M1
Interface: UART

End of Initialization..

Initializing complete.

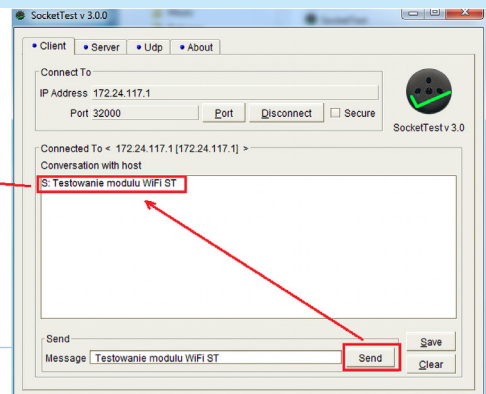
>>setting up miniAP mode...
>>connected...
>>IP address is 172.24.117.1
>>mac addr is 00:80:E1:88:D7:75
```

Rysunek 12. Inicjalizacja modułu – program przykładowy Server-Socket



Rysunek 13. Okno programu SocketTest V3.0.0 pracującego jako klient

```
>>WiFi server socket opening...
>>Server Socket Open OK
...
OClient joined callback...
OClient IP address: 172.24.117.1
OClient MAC address: 00:80:E1:88:D7:75
OClient joined callback...
OClient IP address: 172.24.117.1
OClient MAC address: 00:80:E1:88:D7:75
...
Testowanie modułu WiFi ST
socket fd: 0
msg size: 27
time: size: 27
```



Rysunek 14. Przesłanie tekstu do serwera

Tomasz Jabłoński, EP