

GADALUCH

Sprzętowy klient Gadu-Gadu

AVT-965

Komunikator Gadu-Gadu liczy sobie już 5 lat i ma wielu użytkowników. Zdarza się, że niemal jedynym zadaniem komputera jest obsługa tego programu. Wpływa to na niepotrzebne zużywanie energii elektrycznej i wytwarzanie hałasu, tymczasem rozwiązanie problemu może być bardzo proste.

Rekomendacje:

Gadaluch umożliwia wygodne korzystanie z komunikatora Gadu-Gadu bez konieczności włączania komputera. Jego użytkownicy z pewnością odkryją praktyczne zalety urządzenia.



PODSTAWOWE PARAMETRY

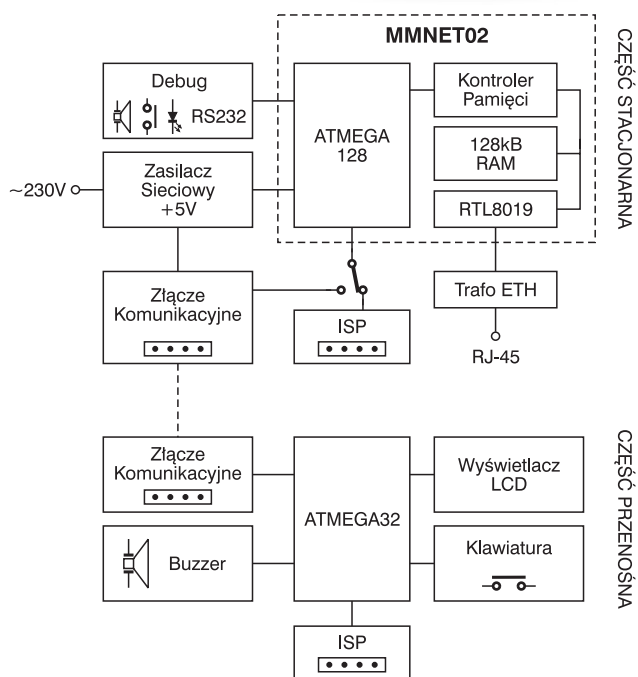
- Płytki o wymiarach: 63x157 mm, 83x109 mm, 87x86 mm
- Zasilanie: 230 VAC (część stacjonarna), +5 VDC (część przenośna)
- Interfejs internetowy: MMNET02
- Wyświetlacz graficzny o rozdzielczości 128x64
- Klawiatura alfanumeryczna w układzie matrycy 7x6 (39 przycisków)
- Sygnalizacja ważnych zdarzeń sygnałem dźwiękowym
- System operacyjny w stacji bazowej: EtherNut

Budowa urządzenia

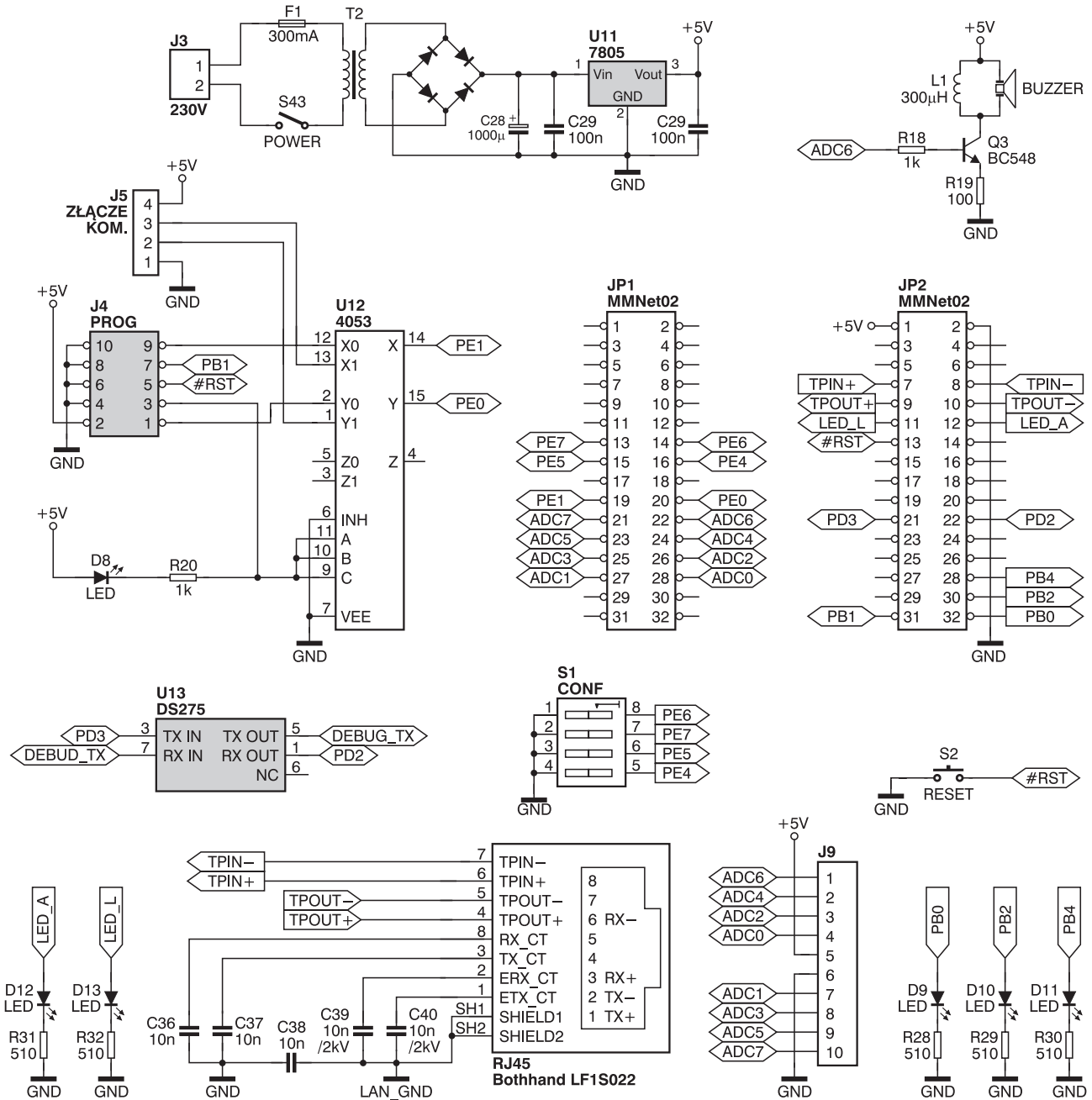
Gadaluch to małe, przenośne urządzenie, posiadające wszystkie podstawowe funkcje, jakie „udostępnia” oryginalny klient Gadu-Gadu. Można w nim wyróżnić część stacjonarną i przenośną. Moduł stacjonarny stanowi „serce” systemu. To w nim znajduje się główny mikrokontroler zarządzający pracą całego urządzenia, interfejs sieci Ethernet oraz szereg pozostałych peryferii. Moduł przenośny jest zdalnym interfejsem użytkownika, jego podstawowym zadaniem jest interakcja z człowiekiem. Dla zapewnienia wygodnej i intuicyjnej obsługi moduł ten wyposażono w duży, czytelny wyświetlacz i klawiaturę alfanumeryczną. Obydwa bloki komunikują się za pomocą portów szeregowych połączonych długim przewodem. Na rys. 1 przedstawiono schemat blokowy prezentowanego urządzenia.

Cześć stacjonarna

Ta część Gadalucha została wykonana w oparciu o moduł internetowy MMNET02 firmy Propox. Jest to niewielka płytk drukowana zawierająca wszystko co potrzebne do zrealizowania połączenia z siecią Ethernet. Ponadto zawiera także wydajny mikrokontroler ATmega128, 128 kB pamięci RAM oraz do 8 MB pamięci Flash (montowanej na życzenie klienta). Wykorzystanie gotowego modułu jest o tyle korzystne, że zwalnia konstruktora z konieczności wykonania i zmontowania płytki z wieloma elementami SMD o niewdzięcznych obudowach – raster wprowadzeń nawet



Rys. 1. Schemat blokowy Gadalucha



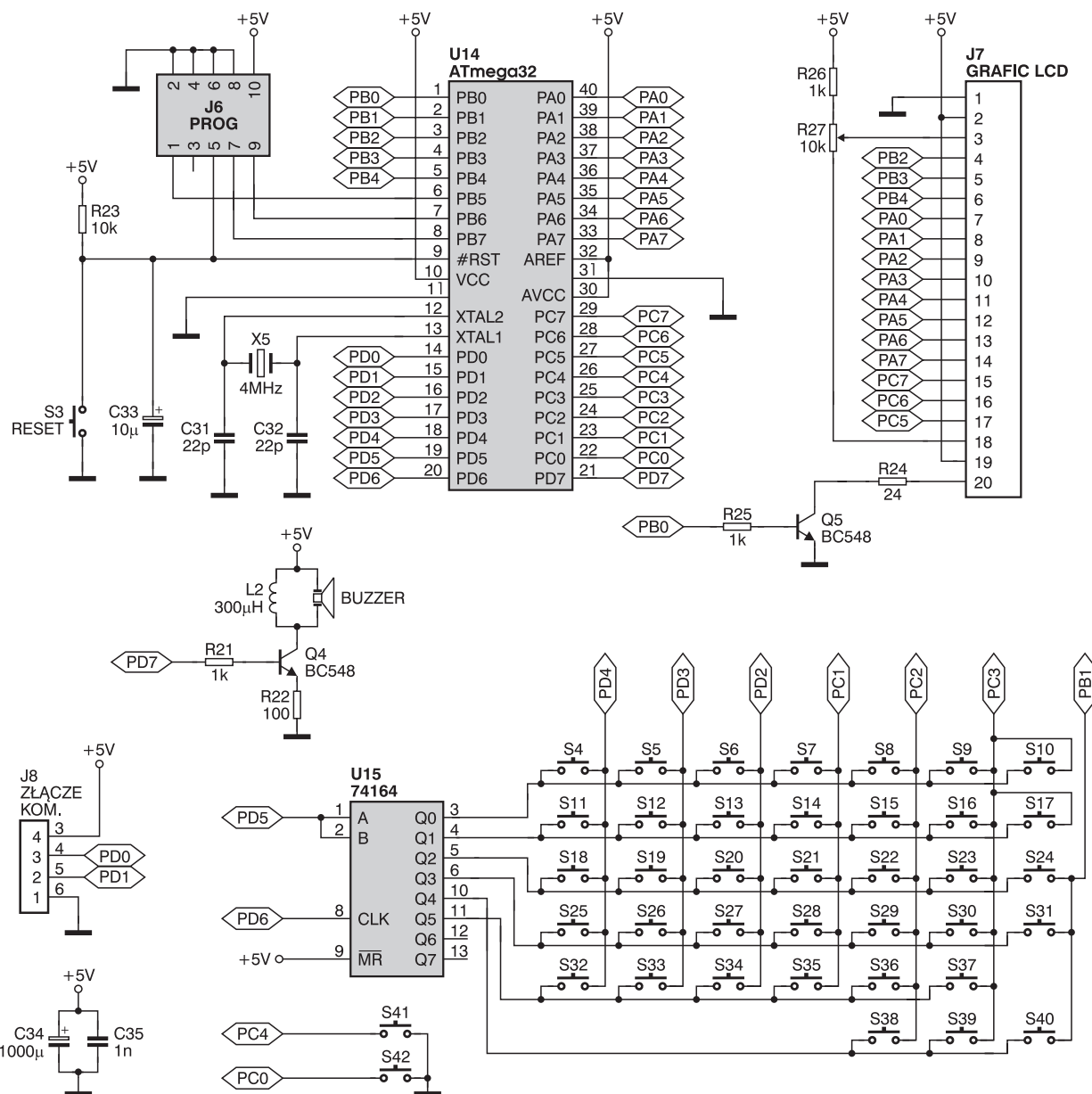
Rys. 2. Schemat elektryczny części stacjonarnej

0,5 mm. Do tego modułu dodano kilka elementów zapewniających zasilanie i kontakt ze światem zewnętrznym – całość przedstawiono na rys. 2.

Na szczególne wyjaśnienie zasługują pewne bloki widoczne na schemacie. I tak, uwagę zwraca obecność elementów nie pełniących żadnych funkcji podczas normalnego użytkowania. Są to: dip-switch (S1), buzzer ze wzmacniaczem (Q3), gniazdo J9, diody LED (D9...D11) oraz układ U13 umożliwiający komunikację szeregową z komputerem. Elementy te miały zastosowanie podczas długiego i żmudnego pro-

cesu uruchamiania urządzenia. Nie muszą być montowane, jeśli urządzenie ma być użytkowane jako komunikator. Zostały umieszczone na wypadek, gdyby ktoś z Czytelników zapragnął zająć się rozwojem oprogramowania zarządzającego pracą Gadalucha. Warto także przyjrzeć się układowi U12. Jest to zespół kluczy analogowych, który w tym układzie pełni rolę przełącznika SPI/USART. Zastosowany mikrokontroler zawiera wiele użytecznych peryferii i mimo aż 64 wyprowadzeń, część z nich pełni dwie, a nawet więcej funkcji. Ma to miejsce także w przypadku pinów E.4 i E.5. Korzysta z nich za-

równy interfejs SPI (używany przy programowaniu pamięci procesora), jak i interfejs USART wykorzystywany podczas normalnej pracy do komunikacji z częścią przenośną prezentowanego urządzenia. Obydwa interfejsy szeregowo są wykorzystywane w układzie. Pracują one jednak na zmianę – nie ma sytuacji, kiedy potrzebne są obydwa naraz. I właśnie do rozwiązania problemu współdzielonych wyprowadzeń wykorzystano wspomniany wcześniej układ U12. Podczas normalnej pracy, do pinów E.4 i E.5 podłączona jest część przenośna urządzenia, natomiast podczas programowania,



Rys. 3. Schemat elektryczny części przenośnej

złącze programatora. Do przełączania wykorzystano sygnał LED dostarczany z programatora – stan niski w czasie programowania. Proces programowania jest sygnalizowany także zapaleniem diody D8.

Cześć przenośna

Cześć przenośna Gadalucha stanowi osobny moduł wyposażony we własny mikrokontroler – ATmega32. Jego głównym, a właściwie jedynym zadaniem jest prowadzenie interakcji z użytkownikiem. Aby zapewnić komfortową i intuicyjną obsługę urządzenia zastosowano duży wyświetlacz graficzny o rozdzielczości 128x64. Jest to wyświetlacz oznaczony symbolem LGM12064ESL, zbudowany w oparciu o sterownik KS0108.

Może on wyświetlić znacznie więcej tekstu niż jego alfanumeryczny odpowiednik o zbliżonych wymiarach. Dodatkowo zastosowany wyświetlacz posiada zielonkawe podświetlenie podnoszące kontrast w trudnych warunkach oświetleniowych. Aby wprowadzanie długich wiadomości nie było uciążliwe, użytkownik otrzymał do dyspozycji klawiaturę alfanumeryczną oraz dodatkowe przyciski rozszerzające jej możliwości i przyspieszające obsługę. Od strony elektronicznej, klawiatura to matryca przycisków w układzie 7x6. By zmniejszyć liczbę linii mikrokontrolera niezbędnych do jej obsługi posłużono się rejestrem przesuwającym 74164 (U15). Za jego pomocą zrealizowano układ „pływającego zera”.

W ten sposób obsługę 39 przycisków (S4...S42) zrealizowano zajmując jedynie 11 pinów procesora. Ważne zdarzenia są sygnalizowane (oprócz komunikatu na wyświetlaczu) w postaci krótkich sekwencji dźwiękowych – zbliżonych do tych, które znamy z oryginalnego komunikatora. Schemat ideowy części przenośnej przedstawiono na rys. 3.

Oprogramowanie

Ważnym elementem każdego systemu wbudowanego jest oprogramowanie. Cześć przenośna Gadalucha została oprogramowana od podstaw. Własnoręcznie stworzono biblioteki zapewniające obsługę peryferii. Mimo to, program jest dość typowy i nie zawiera wielu ciekawych

rozwiązań wartych dokładniejszego omówienia. Inaczej sprawa ma się w przypadku stacjonarnej części urządzenia. Wykorzystano darmowy system operacyjny EtherNut. Jest to dedykowany do pracy na mikrokontrolerach 8-bitowych zaawansowany system wyposażony w stos TCP/IP oraz pseudo współbieżność. Umożliwia on uruchamianie odrębnych zadań w postaci osobnych wątków. Znacznie ułatwia on zbudowanie aplikacji, w której istnieje potrzeba wykonywania kilku zadań jednocześnie, np. obsługa stosu TCP/IP, łączność przez port szeregowy i realizacja żądań obsługi poleceń użytkownika. Przyjrzyjmy się teraz jak wygląda aplikacja pracująca pod kontrolą systemu EtherNut, dla przykładu zostanie omówiona funkcja odpowiedzialna za zalogowanie użytkownika do sieci Gadu-Gadu.

Proces logowania składa się z kilku etapów. Najpierw klient GG przeprowadza krótką sesję HTTP z hostem `appmsg.gadugadu.pl`. Przed nawiązaniem połączenia należy rozwinąć tę nazwę domenową na adres IP. Dzięki zastosowaniu EtherNuta sprawa sprowadza się do wywołania funkcji:

```
NutDnsGetHostByName („appmsg.gadugadu.pl”);
Kiedy adres jest już znany, należy zadbać o utworzenie „kieszeni” – (ang. sock) oraz dołączenie do niej strumienia danych:
NutTcpConnect(sock, serwer_http_ip, 80);
stream = _fdopen(sock, „r+b”);
```

Po nawiązaniu połączenia klient wysyła dane niezbędne do połączenia – w tym numer użytkownika zwany UIN (*User Identification Number*).

WYKAZ ELEMENTÓW

Część stacjonarna

(* – elementy opcjonalne)

Rezystory

- R18*, R20: 1 kΩ
- R19* 100 Ω
- R28*, R29*, R30*: 510 Ω
- R31, R32: 510 Ω

Kondensatory

- C29, C30: 100 nF
- C28: 1000 μF/25 V
- C36...C38: 10 nF
- C39...C40: 10 nF/2 kV

Półprzewodniki

- U12: 4053
- U11: 7805
- Q3*: BC548
- U13*: DS275
- D9*, D10*, D11*: dioda LED
- D8, D12, D13: LED
- D7: mostek prostowniczy 1 A

Inne

- J3: łączówka ARK
- F1: bezpiecznik 300 mA
- L1*: dławik 300 μH
- S1*: dip switch
- JP1, JP2: złącze szpilkowe żeńskie
- S43: włącznik sieciowy
- J4: złącze IDC męskie 2x5 pin

- S2: mikroprzełącznik
- J5, J9*: goldpin
- T2: trafo 12 V
- RJ45 LF1S022 gniazdo z trafram
- Moduł internetowy MMNET02
- Buzzer*

Część przenośna

Rezystory

- R21, R25, R26: 1 kΩ
- R23, R27: 10 kΩ
- R24: 24 Ω
- R22: 100 Ω

Kondensatory

- C31: 33pF
- C35: 1 nF
- C33: 10 μF/25 V
- C32: 33 pF
- C34: 1000 μF/25 V

Półprzewodniki

- U15: 74164
- U14: ATmega32
- Q4, Q5: BC548

Inne

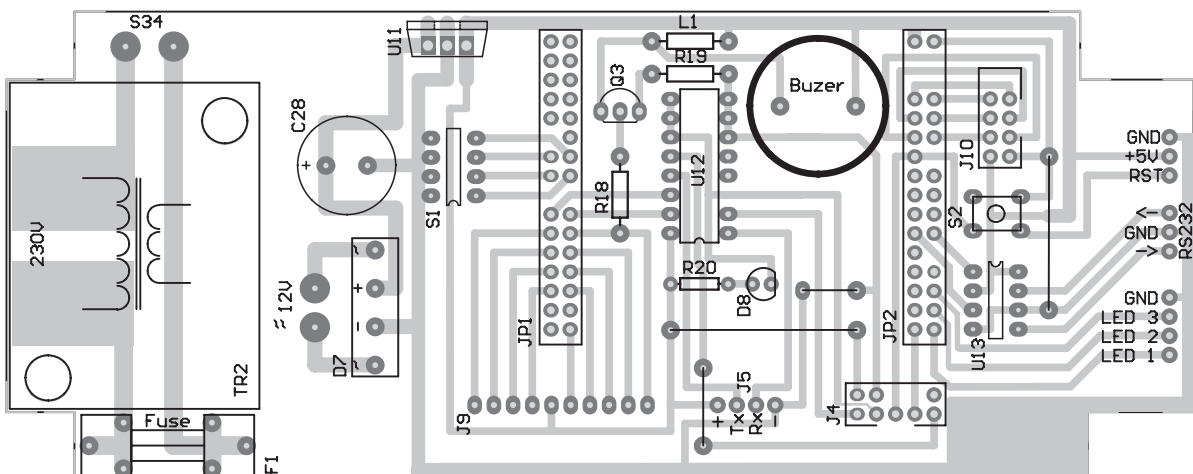
- X5: kwarc 4 MHz
- L2: 300 μH
- J7, J8: łączówki Goldpin
- J6: złącze IDC męskie 2x5 pin
- S3...S4: mikroprzełączniki
- wyświetlacz graficzny LGM12064ESL

```
fprintf(stream, "GET/appsvc/
appmsg4.asp?fmnumber= %lu
&version=6%%2C+1%%2C+0%%2C
+156&fmt=2&lastmsg=1905HTTP/
1.0\r\n""Host:appmsg.
gadugadu.pl\r\n""User-Agent:
Mozilla/4.04 [en] (Win95;
I;Nav)\r\n""Pragma: nocache\
\r\n""\r\n",GGzUser.uin);
fflush(stream);
```

W odpowiedzi otrzymuje adres serwera Gadu-Gadu, z którym należy się łączyć. Poniżej przedstawiono kluczowy fragment sesji (pogrubiono UIN logującego się).

```
(dane wysyłane)
GET/appsvc/appmsg4.asp?fmnumber=3932214&version=6.2.2.15
6&fmt=2
&lastmsg=1905 HTTP/1.0
Host: appmsg.gadugadu.pl
User-Agent: Mozilla/4.04
[en] (Win95; I ;Nav)
Pragma: no-cache\r\n
```

```
(odpowiedź serwera -
pogrubiono IP serwera)
HTTP/1.0 200 OK
0 0 217.17.45.143 217.17.45.
143:8074 217.17.45.143
```



Rys. 4. Schemat montażowy płytki bazowej

