

Andrzej Gorzela
Wydział Informatyki i Zarządzania
Wyższej Szkoły Informatyki w Łodzi

Promotor: dr Dariusz Doliwa

MINIATUROWY SERWER DHCP

Streszczenie – Artykuł opisuje koncepcję miniaturowego serwera DHCP potrafiącego samodzielnie gospodarować adresami IP w lokalnej sieci ethernetowej. Implementację ograniczono do podstawowych funkcji, umożliwiających jednak prawidłową współpracę urządzeń sieciowych z opisywanym serwerem i dystrybucję adresów IP do adresatów.

1 Wstęp

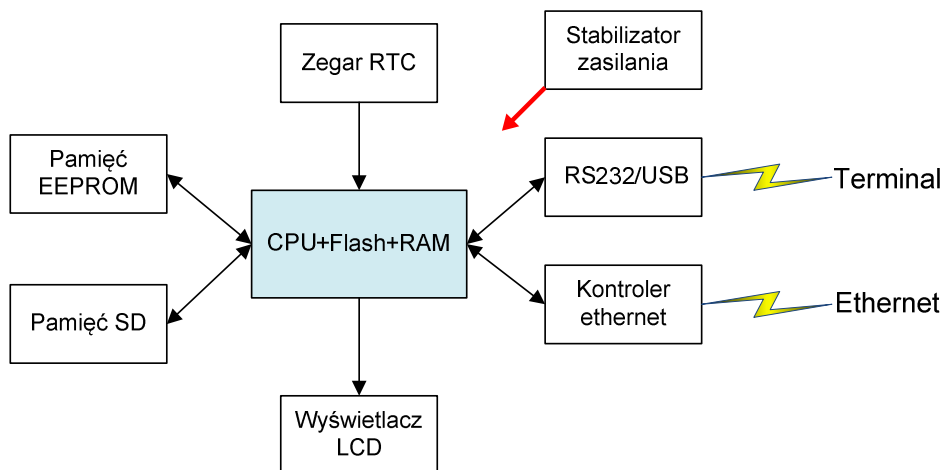
Tematem artykułu jest miniaturowy serwer DHCP. Słowo miniaturowy w tym przypadku znaczy nie mniej, nie więcej jak nie większy od pudełka zapalek. Zastosowanie specjalizowanych układów scalonych, które z roku na rok „potrafią” co raz więcej umożliwić miniaturyzację urządzeń, które jeszcze kilka lat temu potrzebowały co najmniej kilkanaście razy więcej miejsca i konsumowały kilkanaście razy więcej energii oraz kojarzyły się raczej z szafą postawioną gdzieś w serwerowni. Bazując na wiedzy zdobytej podczas studiowania w Wyższej Szkole Informatyki w Łodzi, początkowo chciałem oprzeć się w całości na poprzednim rozwiązaniu i skupić się tylko na ewaluacji i rozwijaniu samego oprogramowania, korzystając ze zbudowanego już osprzętu. Niemniej jednak zamiłowanie do elektroniki nie pozwoliło mi przejść obojętnie obok tematu sprzętu i tak, oprócz nowego oprogramowania powstał także nowy sprzęt, ulepszony, mniejszy, z nowym i szybszym kontrolerem ethernetowym.

2 Projekt systemu

Na rysunku przedstawiono schemat blokowy opisywanego miniaturowego serwera DHCP. Projekt składa się z kilku bloków funkcjonalnych.

- jednostki centralnej CPU wraz z pamięcią ROM (flash) o wielkości 128 kB oraz pamięcią RAM o wielkości 4 kB
- kontrolera ethernet’owego 100 Mb/s

- pamięci masowej w postaci karty SD
- pamięci nieulotnej EEPROM
- konwertera RS232<->USB
- zegara czasu rzeczywistego
- wyświetlacza LCD do komunikacji lokalnej z użytkownikiem/administratorem
- stabilizatora zasilania



Rys. 1. Schemat blokowy serwera

3 Opis funkcjonalny układu

Cały układ serwera zasilany jest z dowolnego źródła napięcia stałego 9-12V. Stabilizator U2 obniża napięcie zasilania do 3,3V. Ponieważ interfejs sieciowy wykonany jest w standardzie 3,3V konieczne było zastosowanie mikrokontrolera w wersji L wykonanej także w standardzie 3,3V. Dzięki temu odpadła konieczność stosowania translatorów poziomów logicznych 5V/3V pomiędzy CPU a resztą układów, co bardzo uprościło to konstrukcję. Co prawda maksymalna częstotliwość taktowania procesora „L” jest dwukrotnie mniejsza od standardowej wersji 5V (16 MHz), ale w tym konkretnym przypadku wydajność samej jednostki centralnej nie jest krytyczna. Zresztą nawet wersja „L” taktowana zegarem 8 MHz zapewnia wystarczającą wydajność serwera. Na rysunku 7 przedstawiono schemat blokowy mikrokontrolera ATMEGA 128L.

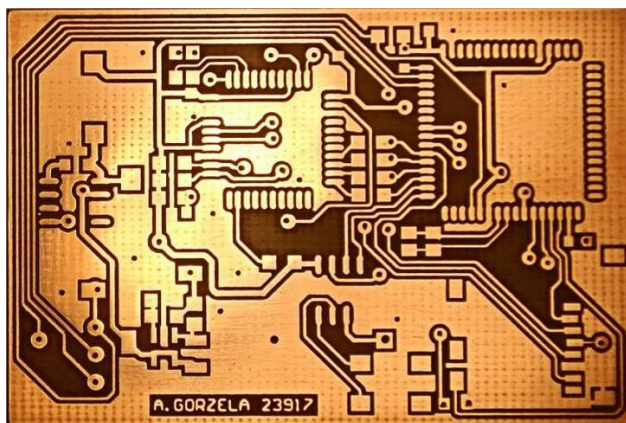
Zasilanie układów cyfrowych stabilizowane jest przez stabilizator impulsowy U2. Zasilanie 12V podświetlenia opcjonalnego wyświetlacza

LCD uzyskiwane jest z przetwornicy impulsowej pracującej na bazie stabilizatora step-up U6. Dzięki zastosowaniu stabilizatorów impulsowych układ zasilania ma wysoką sprawność i minimalne straty mocy, dzięki czemu ciepło wydzielane podczas pracy jest znikome, co ważne jest ze względu na małą konstrukcję urządzenia. Obwód resetowania mikrokontrolera zrealizowano w klasyczny i najprostszy sposób. W czasie włączenia zasilania kondensator C5 ładuje się i tym samym powoduje „ściągnięcie” na chwilę do zera potencjału końcówki RESET, aż do momentu naładowania się kondensatora. Po naładowaniu się kondensatora pin RESET przyjmuje stan wysoki i jest podtrzymywany dalej w tym stanie dzięki zastosowaniu rezystora podciągającego R8. Dodatkowo, dla wygody zamontowano przełącznik resetujący SW1 do ręcznego resetowania serwera. Do interfejsu SPI podłączono kartę SD, pełniącą rolę pamięci masowej, kontroler Ethernetu oraz gniazdo dla opcjonalnego wyświetlacza LCD. Ponieważ nie jest możliwa jednoczesna komunikacja z trzema urządzeniami na raz, procesor sprzęgnięty jest z liniami zezwoleń LCD_CS, ETH_CS oraz SD_CS. Pojawienie się stanu niskiego na linii LCD_CS, SD_CD lub ETH_SD powoduje przejście odpowiedniego urządzenia podpiętego do magistrali SPI w stan aktywny podczas gdy pozostałe urządzenia pozostają „przezroczyste” dla danych. Porty PB5 oraz PB6 wykorzystane są do utworzenia magistrali I2C użytej do komunikacji z zegarem czasu rzeczywistego U3. Podtrzymanie działania zegara po wyłączeniu zasilania realizowane jest za pomocą superkondensatora C3 o pojemności 0,47F. Prawidłowe działanie serwera sygnalizowane jest użytkownikowi poprzez miganie z częstotliwością 1 Hz diody ALIVE podłączonej do portu PD6. Kontroler Ethernetu podłączony jest do magistrali SPI poprzez rezystory 100Ω, które redukują możliwość powstawania „szpilek” zakłócających na magistrali SPI w czasie wymiany danych. W czasie budowy prototypu, kiedy cały serwer zbudowany był na płycie testowej okazało się, że brak tych rezystorów powoduje powstawanie dużej ilości błędów komunikacyjnych pomiędzy CPU a kontrolerem Ethernetu. Pamięć flash mikrokontrolera programować można poprzez złącze P3. Złącze P2 służy do podłączenia wyświetlacza LCD LS020. W poniższej tabeli przedstawiono zestawienie użytych wyprowadzeń mikrokontrolera wraz ich oznaczeniami symbolicznymi zgodnymi ze schematem ideowym oraz opisem pełnionej funkcji.

4 Wykonanie urządzenia

Układ sprzętowy serwera zaprojektowano za pomocą oprogramowania CAD Altium Designer 10. Pozwala on na wykonanie kompleksowego projektu urządzenia elektronicznego, począwszy od

schematu elektrycznego, a kończąc na rysunkach wykonawczych dla wytwórni obwodu drukowanego i szczegółowej liście użytych podzespołów. Ponieważ jednym z założeń, było zbudowanie serwera DHCP wielkości pudełka zapatek wszystkie elementy wykorzystane do budowy serwera wykonane są w technologii montażu powierzchniowego. Największym problemem przy miniaturyzacji były rozmiary gniazda RJ45 wynikające ze standardu wtyku RJ45. Jednak mimo jego znacznych jego rozmiarów w stosunku do pozostałych elementów i tak udało się skonstruować naprawdę niewielkie urządzenie. Na zdjęciu przedstawiono komplet podzespołów przed wykonaniem serwera. Dla lepszego uzmysłwienia rozmiarów niektórych komponentów umieściłem także zapatkę. Montaż serwera wymagał naprawdę zegarmistrzowskiej precyzji. Poniżej umieszczono kilka zdjęć i rysunków z praz związanych z budową serwera. Jako pierwsze widoczne są rysunki projektu z programu Altium Designer. Rozmiar najcieńszej ścieżki wynosi 0,2 mm, a więc mozaika ścieżek jest naprawdę delikatna. Płytką drukowaną wykonana została metodą druku dwustronnego na laminacie 1,5mm.

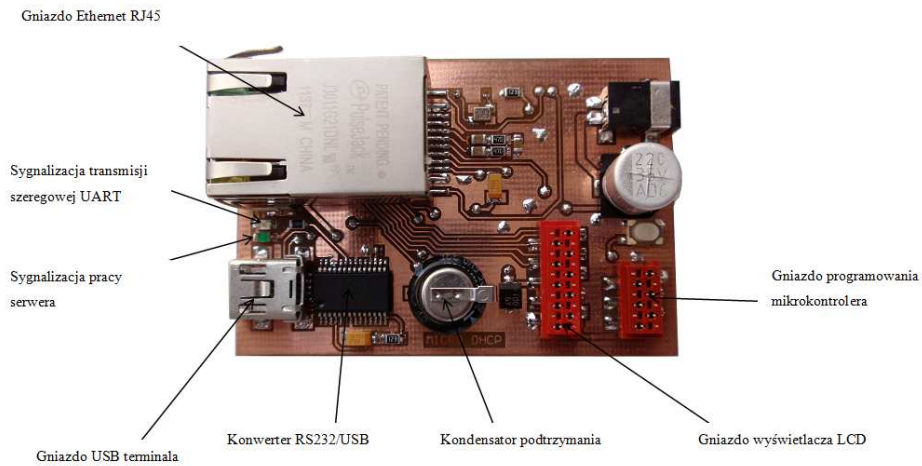


Rys. 2. Widok gotowej, już wytrawionej płytki

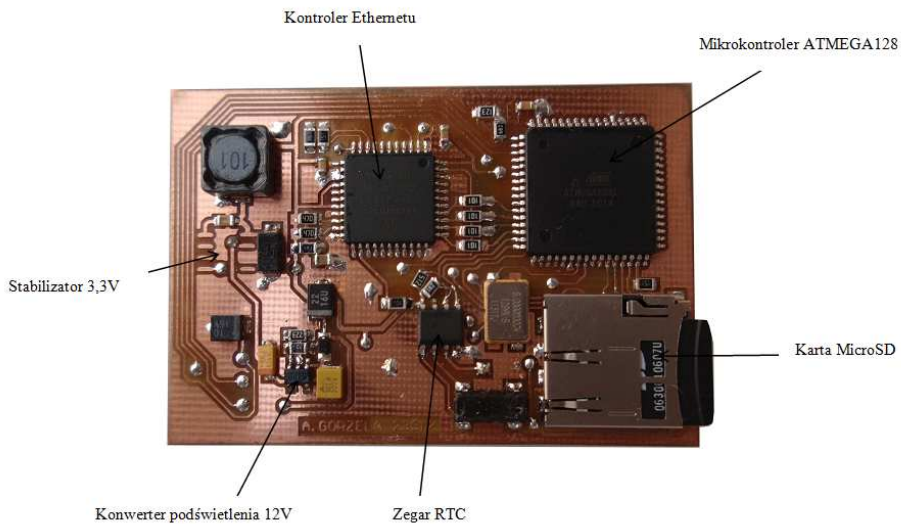
Montaż serwera rozpoczęto od wykonania przelotek pomiędzy warstwami płytki drukowanej.

W następnej kolejności zamontowano elementy stabilizatora 3,3V i wykonano test jego poprawnej pracy. Po upewnieniu się, że stabilizator pracuje poprawnie przystąpiono do montażu elementów biernych – rezystorów i kondensatorów SMD. Po zamontowaniu drobnych elementów przystąpiono do montażu mikrokontrolera ATMEGA oraz pozostałych układów scalonych. Na zakończenie zamontowano

największe elementy – złącze karty SD, kondensator podtrzymania oraz na koniec gniazdo RJ45 wraz z transformatorem separującym.



Rys. 3. Zmontowany serwer - strona górna



Rys. 4. Zmontowany serwer - strona dolna

5 Implementacja programowa serwera

Funkcjonalnie oprogramowanie podzielono na kilka segmentów:

- niskopoziomowa obsługa kontrolera ethernetowego
- stos TCP/IP (ARP, IP, UDP, DHCP, TFTP)

- niskopoziomowa obsługa pamięci SD
- obsługa FAT32
- obsługa zegara RTC
- obsługa wyświetlacza LCD
- parser interfejsu użytkownika
- parser pliku konfiguracyjnego

Jak wspominałem już wcześniej większość elementów programowych stosu TCP/IP oparte są na rozwiązaniu firmy Microchip. Skupiono się na stworzeniu od podstaw samego modułu DHCP i modułów pomocniczych - parsera pliku konfiguracyjnego, parsera interfejsu użytkownika (obsługa terminala) i obsługi wyświetlacza LCD.

Podstawą działania serwera opisywanego serwera DHCP jest blok DCB (DHCP CONTROL BLOCK) w którym zapisywane są dane hostów, ich adresy MAC, przypisane im adresy IP oraz wszelkie informacje związane z dzierżawą adresu IP. Każdy host posiada jeden wpis (slot) w bloku DCB. Każdy slot w DCB znajdować się może w różnych stanach zależnych od stanu dzierżawy adresu:

- DCB_UNUSED – dzierżawa nieużywana (wolny slot)
- LEASE_RELEASED – dzierżawa zwolniona, ale zarezerwowana (host używał już tego adresu, ale nie odświeżył dzierżawy w wymaganym czasie, np. na skutek wyłączenia komputera. Serwer utrzymuje dzierżawę w stanie wstępnej rezerwacji. Nie jest to stan trwały i po określonym czasie wstępna rezerwacja jest kasowana do stanu DCB_UNUSED. Wyjątkiem jest dzierżawa statyczna, która nie kasuje się nigdy. Zapobiega to zapchaniu się bloku DCB, kiedy to kilkunastu klientów zażądałoby rezerwacji adresu, a potem na dłuższy czas zniknęło z sieci. Nowi klienci nie byłiby w stanie otrzymać rezerwacji ze względu na przepełnienie bloku DCB. W momencie kiedy host pojawi się w sieci z powrotem, serwer zaproponuje hostowi wcześniej używany adres.
- LEASE_DECLINED – dzierżawa zabroniona. W przypadku kiedy serwer zaproponuje adres IP, który zapisany jest w tablicy ARP hosta jako należący do innego hosta (serwer może nie wiedzieć o istnieniu takiego hosta w momencie wysyłania propozycji dzierżawy) host odpowiada serwerowi wiadomością DHCP_DECLINE. Serwer oznacza ten adres IP jako zabroniony.
- LEASE_REQUESTED – host zażądał od serwera wydzierżawienia adresu IP za pomocą wiadomości DHCP_REQUEST
- LEASE_GRANTED – adres IP jest w stanie dzierżawy

- ARP_REQUESTED - serwer wysłał zapytanie ARP do sieci w celu identyfikacji czy adres przeznaczony do dzierżawy nie jest już używany przez nieznanego przez serwer hosta.

6 Inicjalizacja systemu – restart

System przed rozpoczęciem normalnej pracy musi być odpowiednio skonfigurowany.

Po restarcie wykonywany jest szereg funkcji “podnoszących” kolejno odpowiednie fragmenty sprzętu. Pierwszymi operacjami wykonywanymi po restarcie serwera są komendy związane z konfiguracją portów mikrokontrolera, przerwań sprzętowych oraz konfiguracją timerów systemowych. Są to standardowe czynności i nie wymagają szczegółowych wyjaśnień. Po inicjalizacji przerwań, portów oraz timerów system kolejno :

- uruchamia magistralę I²C za pomocą funkcji `i2c_init()`. Magistrala służy do komunikacji z zegarem czasu rzeczywistego M41T81.
- uruchamia zegar czasu rzeczywistego za pomocą funkcji `rtc_init()`
- konfiguruje złącze szeregowo za pomocą funkcji `UARTInit()`. Parametry portu szeregowego ustawiono na 38400,8,1,N.
- uruchamia port SPI za pomocą funkcji `ConfigureSPIModule()`. Port SPI obsługuje komunikację z interfejsem sieciowym, kartą SD oraz opcjonalnie z wyświetlaczem LCD.
- uruchamia programowy licznik czasowy za pomocą funkcji `TickInit()`.
- inicjalizuje dysk SD za pomocą funkcji `disk_initialize()` oraz `mount()`
- kopiuje z pamięci nieulotnej dane konfiguracyjne i umieszcza je w zmiennych procesowych za pomocą funkcji `InitAppConfig()`
- inicjalizuje stos TCP/IP za pomocą funkcji `StackInit()`
- startuje serwer DHCP za pomocą funkcji `DHCPServerInit()`
- inicjalizuje parser linii komend i wyświetla znak zachęty na podłączonym terminalu za pomocą funkcji `CommandLineInit()` oraz `CommandPrompt()`

7 Podsumowanie i wnioski

Budowa miniaturowego serwera DHCP okazała się doskonałym sposobem na poznanie kolejnego, powszechnie używanego protokołu sieciowego. Jak się okazało konstrukcja protokołu DHCP jest dość

prosta i implementacja go nie przysporzyła większych problemów. Zakończenie budowy urządzenia pozwoliło wysnuć kilka wniosków. Praca serwera okazała się zaskakująco sprawna. Urządzenie działa stabilnie i w czasie kilkunastodniowych testów nie sprawiało żadnych problemów. Podsumowując aspekty konstrukcyjne można stwierdzić, że konstrukcję można było uprościć przez rezygnację z karty pamięci SD na rzecz zapisu pliku konfiguracyjnego w pamięci EEPROM. Dodatkowo zmniejszyłoby to kod programu, gdyż nie trzeba by implementować obsługi FAT16, a plik przechowywać w pamięci EEPROM w postaci obrazu binarnego. Początkowo zamierzano używać serwera także jako małego serwera plików, jednakże rezygnacja z tego zamierzenia także postawiła pod znakiem zapytania sens stosowania karty SD jako medium przechowywania bądź co bądź niewielkiego pliku konfiguracyjnego. Ale urządzenie jest rozwojowe i nic nie stoi na przeszkodzie, aby wykorzystać je do wspomnianego celu.

Ponieważ tablica dzierżaw przechowywana jest w pamięci RAM mikrokontrolera, której wielkość jest ograniczona (odejmując 1 kB potrzebny na pracę stosu oraz kilkaset bajtów na zmienne systemowe zostaje niecałe 3 kB na tablice dzierżawy) serwer pozwala na obsłużenie kilkunastu hostów. Można by pokusić się o zapis tablicy dzierżaw na karcie SD, co oczywiście jest jak najbardziej możliwe, lecz w założeniu urządzenie miało mieć bardziej wartość edukacyjną niż użytkową więc ilość kilkunastu dzierżaw na raz jest moim zdaniem w tym wypadku wystarczająca.

Można by pokusić się o wbudowanie w urządzenie małego serwera WWW umożliwiającego kontrolę pracy poprzez przeglądarkę internetową i interfejs graficzny. Ułatwiłoby to konfigurację i monitoring pracy urządzenia. Wzrosłyby oczywiście wymagania dotyczące sprzętu i wskazana byłaby wymiana mikrokontrolera na inny, mający większą pamięć RAM. Mimo braku tego udogodnienia, uważam, że obecny sposób konfiguracji i tak jest stosunkowo prosty.

Budowane urządzenie mogłoby doskonale sprawdzić się przy budowie instalacji inteligentnego budynku, gdzie poszczególne podzespoły instalacji (wyświetlacze, moduły wykonawcze itp.) połączone byłyby między sobą za pomocą dedykowanej sieci Ethernet. Gospodarkę adresową IP w takiej sieci prowadzić mógłby opisany serwer.

Niniejsza praca jest doskonałym przykładem, jak urządzenia sieciowe kojarzące się kiedyś z serwerownią i szafami wypełnionymi kilogramami sprzętu „kurczą się” i trafiają w świat mikrokontrolerów i systemów „zaszytych”. Wzrastające możliwości mikrokontrolerów, wzrost prędkości ich działania oraz zasobów pamięciowych i sprzętowych sprawiają, że Ethernet staje się standardem komunikacyjnym także tutaj. Powstają swoiste mikrosieci. Mikro, ale rządzące się tymi samymi zasadami.

Pojawianie się na rynku elektroniki coraz to nowszych scalonych kontrolerów ethernetowych powoduje, że do Internetu można podłączyć praktycznie każde urządzenie elektroniczne wyposażone w mikroprocesor. Pralka podłączona do Internetu ? Czemu nie ? W końcu mamy 21 wiek.

Literatura

- [1] Droms R. E., Lemon T., *The DHCP Handbook*, 2002
- [2] Alcott N., *DHCP for Windows 2000*, 2002
- [3] Karanjit S. S., Parker T., *TCP/IP. Księga eksperta*. Wydanie II, 2002
- [4] Francuz T., *Język C dla mikrokontrolerów AVR. Od podstaw do zaawansowanych aplikacji*, 2011
- [5] RFC 2131 Dynamic Host Configuration Protocol 1997
- [6] RFC 1533 - DHCP Options and BOOTP Vendor Extensions 1993
- [7] RFC 951 BOOTSTRAP PROTOCOL (BOOTP) 1983
- [8] AN833 Microchip TCP/IP Stack Application Note

TINY DHCP SERVER

Summary – The article describes the concept of DHCP server the ability to independently manage a miniature IP addresses in the local Ethernet network. Implementation was limited to basic functions, enabling network device with this cooperation, however, the correct server and distribute the IP addresses.