

Ethernet i AVR-y

Ethernet od podstaw, część 7

Duża część systemów mikroprocesorowych musi mieć dostęp do aktualnej daty i godziny. Niestety wiąże się to z okresowym ustawianiem i korygowaniem wskazań zegara RTC. Jeśli połączymy nasze urządzenie z Internetem, możemy zapomnieć o tej przykryj konieczności. W ostatnim odcinku kursu zajmiemy się zatem funkcjami związanymi z czasem w systemie Nut/OS.

Czas w systemie Nut/OS

Jednym z powodów stosowania systemów operacyjnych na mikrokontrolerach jest uniezależnienie (w pewnym stopniu) oprogramowania od sprzętu. Oznacza to, że niektóre peryferia programuje się identycznie na wszystkich architekturach sprzętowych obsługiwanych przez system. W Nut/OS przykładem takiej niezależności są funkcje obsługujące zegar czasu rzeczywistego. Ponadto są one zgodne z ich „pecetowymi” odpowiednikami znajdującymi się w standardowej bibliotece języka C.

Zacznijmy od pobierania aktualnej daty i godziny – wykonuje się to w następujący sposób:

```
time_t t_sec = time(NULL);
struct _tm czas = localtime(&t_sec);
```

Wywoływana w pierwszej linii kodu funkcja `time()` zwraca aktualny czas w postaci liczby sekund od północy dnia 1 stycznia 1970 czasu uniwersalnego (UTC – *Universal Coordinated Time*). Taka reprezentacja daty i godziny jest standardem w systemach uniksowych (POSIX Time), a sam sposób zapisu czasu w postaci liczby cykli, które upłynęły od określonej daty początkowej jest wykorzystywany w niemal wszystkich systemach operacyjnych (np. Windows odmie-

rza czas jako liczbę 100-nanosekundowych cykli od początku roku 1601). Choć takie formaty wydają się nieco dziwne, ułatwiają odmierzenie zadanych odstępów czasu.

Drużga linia kodu przekształca wartość zwróconą przez funkcję `time()` na postać bardziej zrozumiałą dla człowieka – strukturę `_tm`, której składowe opisane są w ramce poniżej. Funkcja `localtime()` zwraca datę i godzinę w lokalnej strefie czasowej. Ponieważ zegar przechowuje czas UTC, konieczne jest ustawienie właściwej strefy czasowej. Do tego celu służy zmienna globalna `_timezone`, zawierająca różnicę w sekundach między lokalną strefą czasową a czasem uniwersalnym. W Polsce jest to -1 godzina, czyli -3600 sekund. Należy pamiętać o poprawnym zainicjowaniu `_timezone` w swoim programie.

System Nut/OS obsługuje również automatyczną zmianę czasu letniego na zimowy i vice versa. Sterują tym dwie kolejne zmienne globalne:

`_daylight` – zapisanie niezerowej wartości powoduje włączenie automatycznej zmiany czasu,

`_dstbias` – różnica między czasem letnim a zimowym w sekundach



Do myślnie ustawienia zmiennych `_daylight`, `_dstbias` są poprawne dla obszaru Polski. Odczytywanie bieżącego czasu mamy opanowane, pora na ustawianie zegara. Polega to na:

– przekształceniu czasu lokalnego ze struktury `czas_do_ustawienia` na czas w formacie POSIX Time za pomocą funkcji `mktime()`:

```
struct _tm czas_do_ustawienia; // struktura z czasem, który ma być ustawiony
time_t t_sec =
mktime(&czas_do_ustawienia);
– ustawieniu zegara za pomocą funkcji stime()
stime(&t_sec);
```

Protokół SNTP i synchronizacja zegarów

Na wstępie wspomniałem, że jeśli połączymy nasze urządzenie z Internetem, możemy zapomnieć o ręcznym ustawianiu zegara. Jest to możliwe dzięki protokołom sieciowym NTP (*Network Time Protocol*) i jego uproszczonej wersji SNTP (*Simple Network Time Protocol*), którego obsługa jest wbudowana w system Nut/OS. Serwery SNTP nasłuchują na porcie 123 protokołu UDP. Pobranie aktualnej



Format struktury `tm` przechowującej datę i godzinę:

```
struct _tm {
    int _tm_sec;           – sekundy (0..59)
    int _tm_min;          – minuty (0..59)
    int _tm_hours;        – godziny (0..23)
    int _tm_mday;         – dzień miesiąca (1..31)
    int _tm_mon;          – miesiąc (0 – styczeń, 11 – grudzień)
    int _tm_year;         – rok (aktualna wartość jest równa 1900 + tm_year)
    int _tm_wday;         – dzień tygodnia (0 – niedziela)
    int _tm_yday;         – dzień roku
    int _tm_isdst;        – wartość niezerowa oznacza czas letni
};
```

List. 1. Najistotniejsze fragmenty kodu programu zegara wykorzystującego protokół SNTP (SNTPDemo)

```

int main(void)
{
    initialize();
    init_network();

    // ustawienie strefy czasowej w sekundach (-1 godzina od czasu GMT)
    _timezone = (-1) * 60L * 60L;

    LCD_clear();
    u_long addr = NutDnsGetHostByName(SNTP_SERVER_ADDR);

    NutSNTPStartThread(addr, 30000);          // aktualizujemy zegar co 5 minut

    for(;;)
    {
        char LCD_line1[16], LCD_line2[16];
        time_t t_sec = time(NULL);
        struct tm *t = localtime(&t_sec);
        sprintf(LCD_line1, „Data: %02u.%02u.%u”, t->tm_mday, t->tm_mon + 1, 1900 + t->tm_year);
        sprintf(LCD_line2, „Czas: %02u:%02u:%02u”, t->tm_hour, t->tm_min, t->tm_sec);

        LCD_clear();
        LCD_setpos(0);
        LCD_putstr(LCD_line1);
        LCD_setpos(40);
        LCD_putstr(LCD_line2);

        NutSleep(500);
    }
}

```

List. 2. Zmodyfikowany serwer WWW obsługujący uwierzytelnianie użytkowników (fragmenty kodu przykładu HTTPServerAuth)

```

main()
{
    initialize();
    init_network();
    (...)

    NutRegisterAsp();
    NutRegisterAspCallback(ASP_callback);
    NutRegisterCgi(„ledlcd.cgi”, CGI_callback);

    // zabezpieczamy hasłem katalog cgi-bin zawierający skrypt sterujący diodami LED i wyświetlaczem.
    NutRegisterAuth(„cgi-bin”, „user:haslo”);
    // oraz drugi katalog 'tajne' zawierający stronę ze ściśle tajnymi informacjami :)
    NutRegisterAuth(„tajne”, „user2:haslo2”);

    NutThreadCreate(„lcds”, lcd_scrolling, NULL, 256);

    for(i=0; i<5; i++)
    {
        sprintf(str, „httpd%d”, i);
        NutThreadCreate(str, httpd_thread, NULL, NUT_THREAD_MAINSTACK);
    }

    NutThreadSetPriority(254);

    for(;;) NutSleep(10000);
}

```

daty i godziny polega na wysłaniu pakietu z odpowiednim zapytaniem i odebraniu odpowiedzi zawierającej aktualny czas UTC. Maksymalne odchylenie odebranego czasu od wzorca zależy od opóźnienia transmisji pakietu przez sieć i zwykle nie przekracza kilkuset milisekund. Aby uruchomić synchronizację czasu w systemie Nut/OS, wystarczy wywołać funkcję:

```
int NutSNTPStartThread(u_long server_addr, u_long interval);
```

Tworzy ona osobny wątek okresowo korygujący ustawienia zegara czasu rzeczywistego. Czas jest pobierany z serwera SNTP o adresie `server_addr` co `interval` milisekund.

Lista publicznych serwerów SNTP znajduje się między innymi na stronie <http://ntp.isc.org/bin/view/Servers/WebHome>. Na **list. 1** przedstawiono kod prostej aplikacji zegara nie wymagającego ręcznego ustawiania. Data i godzina jest pokazywana na wyświetlaczu alfanumerycznym 2x16 znaków. Program korzysta z serwera `tempus1.gum.gov.pl` znajdującego się w Głównym Urzędzie Miar i synchronizowanego zegarem atomowym.

Uzupełnienia

Jeden z Czytelników zwrócił uwagę na pominięcie przeze mnie jednej z ważnych funkcji ethernetowego serwera HTTP – uwierzytelniania użytkowników. Pozwala ono na zabezpieczenie dostępu do okre-

ślonych katalogów strony internetowej za pomocą hasła. Aby uruchomić mechanizm uwierzytelniania, należy podczas inicjacji serwera HTTP skorzystać z funkcji:

```
int NutRegisterAuth(CONST char *dirname, CONST char *login).
```

Parametr `dirname` to nazwa zabezpieczonego katalogu, `login` – nazwa użytkownika i hasło rozdzielone dwukropkiem. Hasła są przesyłane przez przeglądarkę do serwera w postaci jawnej i mogą być podsłuchane. Serwer HTTP wbudowany w Nut/OS nie obsługuje połączeń szyfrowanych, dlatego nie powinno się go stosować w aplikacjach wymagających wysokiego poziomu bezpieczeństwa. Zmodyfikowany

kod serwera WWW z czwartej części kursu, w którym zmiana stanu diod LED i napisu na wyświetlaczu wymaga uwierzytelnienia użytkownika (domyślnie – user: user, hasło: hasło) znajduje się na list. 2.

Podsumowanie – co po AVR-ach?

Pierwsza wersja oryginalnego zestawu Ethernut, której odpowiednikiem jest nasz kursowy zestaw

ZL9AVR+ZL1ETH, powstała w 2002 roku. Wówczas w Polsce niepodzielnie królowały układy z rodziny AVR. Od tego czasu minęło 5 lat (w elektronice to cała epoka, albo i dwie...). Dziś 32-bitowe mikrokontrolery z rdzeniami ARM są łatwo dostępne i bardzo tanie, a ich moc obliczeniowa pozwala na realizowanie znacznie ciekawszych aplikacji niż przedstawione w ramach niniejszego kursu. Przykładami mogą

być obiórnik internetowych stacji radiowych oraz radiowęzeł lub system monitoringu wykorzystujący do transmisji danych sieć Ethernet. Niektóre z nich już niebawem zobaczymy na łamach EP.

Życzę powodzenia w realizowaniu własnych aplikacji dla systemu Nut/OS!

Tomasz Włostowski, EP
tomasz.wlostowski@ep.com.pl

WYKRYWACZE METALI

CS150

Dyskryminator audio
VU meter
Wodoszczelna sonda (20 cm)



Cena: 390 zł

Cena: 190 zł

CS10MD

Wykrywacz "ręczny"
Idealny dla policjantów
i ochroniarzy.



Zamówienia przyjmuje Dział Handlowy AVT

01-939 Warszawa, ul. Burleska 9, tel. 022 568 99 50, fax 022 568 99 55, e-mail: handlowy@avt.pl, www.sklep.avt.pl

Prenumerata e-wydania

Elektronikę Praktyczną możesz czytać na monitorze swego komputera w postaci identycznej z wydaniem papierowym!

wbudowane linki

klikasz i jesteś na odpowiedniej stronie WWW

hipertekstowy spis treści i wyszukiwarka

od razu znajdziesz to, czego szukasz

wygodne archiwum

czyli poprzednie wydania pod ręką

multimedia

animacje, dźwięk, wideo



E-prenumeratę można zamawiać (na www.ep.com.pl/eprenumerata lub www.avt.pl/eprenumerata) na 6, 12 lub 24 wydania w cenie odpowiednio: 6,20 zł, 5,70 zł i 5,20 zł za wydanie (patrz str. 141)

2xgratis!

Prenumeratory wydania papierowego otrzymują za darmo również e-prenumeratę
Jedno okazowe **e-wydanie** archiwalne można zamówić za darmo tytułem próby