XX Jubileuszowe Seminarium

ZASTOSOWANIE KOMPUTERÓW W NAUCE I TECHNICE' 2010
Oddział Gdański PTETiS
Referat nr 23

SYSTEM STEROWANIA CZUJNIKAMI Z WYKORZYSTANIEM SIECI BEZPRZEWODOWEJ W STANDARDZIE IEEE 802.15.4

Bogusław PAROL, Lech HASSE

1. Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji I Informatyki
Katedra Optoelektroniki i Systemów Elektronicznych
tel: +48 58 347 1884 fax: +48 58 341 6132 e-mail: lhasse@eti.pg.gda.pl

Streszczenie: Bezprzewodowe sieci sensorowe najczęściej wykorzystywane są w systemach pomiarowo-kontrolnych. Przy realizacji systemu wykorzystano mikrokontroler z rdzeniem Cortex-M3 jako kontroler sieci oraz przekaźniki ZigBee pracujące przy częstotliwości 2,4 GHz. System wymagał stworzenia modułów komunikacyjnych łączących kontroler i przekaźnik radiowy za pomocą interfejsu SPI oraz zaimplementowania stosu odpowiadającego potrzebom aplikacji. Opracowano oprogramowanie sterujące bezprzewodowo pracą urządzeń automatyki za pomocą mikrokontrolera STM32 w oparciu o standard IEEE 802.15.4 oraz kompleksowe, wielofunkcyjne sterowanie systemem z komputera PC.

Słowa kluczowe: sensorowa sieć bezprzewodowa, standard IEEE 802.15.4, protokoły komunikacji

1. WPROWADZENIE

Sensorowe sieci bezprzewodowe są stosowane w systemach kontrolno-pomiarowych rozprzestrzenionych na większym obszarze; dotyczy to systemów automatyki przemysłowej, inteligentnych budynków, zastosowań w instytucjach służby zdrowia, a nawet w systemach bezpieczeństwa czy też w systemach wojskowych. Bogata oferta dostępnych produktów pozwala na łatwą implementację istniejących rozwiązań sprzętowych wewnątrz własnych systemów.

Przy realizacji systemu sterowania bezprzewodowo czujnikami urządzeń automatyki standardzie komunikacyjnym IEEE 802.15.4, przyjęto następujące ustalenia jako zadania cząstkowe:

- zastosowanie mikrokontrolera z rdzeniem Cortex-M3 jako kontrolera sieci oraz przekaźników ZigBee pracujących przy częstotliwości 2,4 GHz,
- stworzenie modułów komunikacyjnych przez połączenie kontrolera i przekaźnika radiowego za pomocą interfejsu SPI (*Serial Peripheral Interface*),
- zaimplementowanie liniowo uporządkowanej struktury danych (stosu) odpowiadającej potrzebom systemu,
- stworzenie oprogramowania sterującego pracą urządzeń automatyki oraz oprogramowania na PC, umożliwiającego łatwe i wielofunkcyjne sterowanie systemem.

Komitet IEEE zakończył pracę nad standardem 802.15.4 (*ZigBee Certified*) w 2003 roku. Jest on obecnie bardzo rozpowszechniony w systemach komunikacji, w których nie jest istotna duża szybkość transmisji, a efektywność i energooszczędność.

2. OPIS FUNKCJONALNY SYSTEMU

Po analizie dostępnych na rynku komponentów wybrano mikrokontroler STM32 z rdzeniem Cortex-M3 STMicroelectronics oraz radio-przekaźnik SPZB260-PRO z zaimplementowanymi warstwami MAC i PHY zgodnymi z najnowszą specyfikacją ZigBee PRO. System został wzbogacony o oprzyrządowanie pozwalające na sterowanie za pomocą klawiatury i wyświetlacza LED koordynatora sieci w systemie operacyjnym RTOS zaimplementowanym w pamięci kontrolera sieci po stronie koordynatora.

Prezentowany system składa się z trzech podstawowych modułów:

- komputera klasy PC z oprogramowaniem (środowisko LabWindows CVI) do sterowania elementami systemu za pomocą jednego z dwóch wspieranych portów komunikacyjnych: USB lub portu szeregowego COM,
- modułu koordynatora sieci bezprzewodowej, zrealizowanego na zestawie uruchomieniowym ZL27ARM, wyposażonego w mikrokontroler STM32F103VBT6 oraz podłączony do niego za pomocą interefejsu SPI moduł ZigBee SPZB260PRO,
- modułu urządzenia końcowego sieci, wyposażonego w mikrokontroler STM32F103RBT6 oraz podłączony do niego za pomocą interfejsu SPI moduł ZigBee SPZB260PRO a także cyfrowego czujnika temperatury DS18B20 oraz czujnika natężenia światła zbudowanego przy wykorzystaniu fototranzystora BPW40.

Program sterujący systemem komunikuje się z modułem koordynatora sieci za pomocą portu szeregowego komputera PC. Mikrokontroler pełniący rolę hosta w module koordynatora wspiera komunikację zarówno przy użyciu interfejsu USB jak RS232, oba te rozwiązania mogą być zaimplementowane w zależności od potrzeb aplikacyjnych

Recenzent: Dr inż. Krzysztof Nyka – Wydział Elektroniki, Telekomunikacji i Informatyki Politechnika Gdańska

systemu. W opisanym rozwiązaniu zastosowano komunikację za pomocą portu USB oraz wirtualnego portu COM zainstalowanego na komputerze PC, dzięki czemu oprogramowanie sterujące pracą systemu staje się uniwersalne i może wspierać oba interfejsy komunikacyjne. Do obsługi interfejsu USB użyto sterowników CDC (*Common Device Driver*) udostępnionych przez producenta wykorzystanych w systemie mikrokontrolerów.

Polecenia przekazywane do mikrokontrolera, podłączonego do PC za pomocą portu USB, są w nim dekodowane a następnie transmitowane do modułu ZigBee za pośrednictwem interfejsu SPI. Moduł urządzenia końcowego sieci odbiera przesłaną drogą bezprzewodową informację, aby następnie przekazać ją do swojego hosta, mikrokontrolera SM32F103RBT6. Dekoduje on polecenia i wyzwala odpowiednią akcję. Informacja zwrotna przesyłana jest do koordynatora sieci, który przekazuje ją do aplikacji sterującej pracą systemu i kończy wymianę danych.

3. ZASTOSOWANE PROTOKOŁY KOMUNIKACJI I ICH ZABEZPIECZENIA

Aby zapewnić poprawną komunikację pomiędzy modułami systemu, wykorzystano procedurę obliczania i dodawania do przesyłanych ramek danych cyklicznego kodu nadmiarowego *CRC*. I tak, w przypadku komunikacji za pośrednictwem interfejsu USB wykorzystano kod nadmiarowy *CRC*8, którego wielomian charakterystyczny ma postać:

$$CRC(x) = x^8 + x^5 + x^4 + 1$$

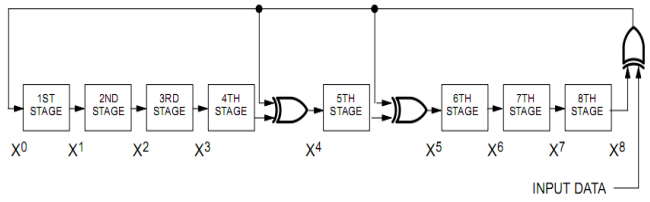
Stworzony protokół transmisji zakłada, że transmitowane ramki danych mają długość 8 bajtów, gdzie kolejne bajty zawierają: kod urządzenia DC (*device code*), kod polecenia CC (command code) oraz parametry lub dodatkowe informacje CPx (*command parameter x*), a na ostatnim miejscu znajduje się obliczona suma kontrolna CRC (*Cyclic Redundancy Code*) (Tab. 1).

Tablica 1. Ramka wykorzystanego protokołu w transmisji USB

DC	CC	CP1	CP2	CP3	CP4	CP5	CRC

Obliczona na podstawie pierwszych siedmiu bajtów suma kontrolna dodawana jest na ostatniej pozycji w ramce i transmitowana do mikrokontrolera, który również oblicza sumę CRC na podstawie odebranch pierwszych siedmiu bajtów, a następnie porównuje ją z wartością ostatniego bajtu.

Zastosowano taką postać wielomianu *CRC8*, ponieważ właśnie taki kod wykorzystuje cyfrowy czujnik temperatury DS18B20. Wbudowany w nim generator CRC do obliczania sumy kontrolnej wykorzystuje rejestr przesuwny (rys.1) [1].



Rysunek 1. Rejestr przesuwny generatora CRC8

Dzięki temu poprawność transmitowanych z czujnika danych może być sprawdzana po stronie mikrokontrolera.

Również w transmisji bezprzewodowej wykorzystana jest nadmiarowa suma kontrolna CRC. Jednostka odpowiedzialną za jej obliczanie, dodawanie do przesyłanego pakietu oraz porównywanie jest zintegrowany, sprzętowy moduł MAC zgodny z wymogami standardu IEEE 802.15.4-2003. Moduł MAC wbudowany jest w procesor sieciowy SN260, który jest główną jednostką układu ZigBee SPZB260PRO.

Dedykowany procesor sieciowy SN260 zawiera stos ZigBee Znet firmy EMBER, który jest w pełni zgodny ze standardem ZigBee PRO (wzbogacony o wiele nowych możliwości profil ZigBee 2007). Sterowanie nim możliwe jest zarówno za pomocą interfejsu UART, jak również przez interfejs SPI. Komunikacja odbywa się za pomocą dedykowanego protokołu EZSP (*Ember Znet Serial Protocol*), w którym stosuje się ramki o zmiennej długości, zależnej od treści przesyłanej wiadomości (tab. 2).

Tablica 2. Ramka protokołu EZSP

SQ	FC	FID	P_0	P_1	...	P_{N-1}

gdzie:

SQ (*sequence*) oznacza numer wiadomości; host powinien inkrementować wartość mieszczącą się w tym polu za każdym razem, gdy wysyła wiadomość. Odpowiedź zwrotna również zawiera to pole, dzięki czemu dana wiadomość może zostać odpowiednio zakwalifikowana.

FC (*frame control*) określa rodzaj przesyłanej wiadomości. Najstarszy bit determinuje, czy dana wiadomość jest poleceniem czy też odpowiedzią na wcześniej otrzymane polecenie, a najmłodsze dwa bity odpowiadają za tryb pracy modułu ZigBee (ustawienie tych bitów decyduje o zmianie trybu pracy na jeden z możliwych: *power down*, *deep sleep*, *idle*),

FID (*frame ID*) zawiera kod polecenia,

$P_0 \div P_{N-1}$ zawierają parametry danego polecenia lub treść przesyłanej wiadomości.

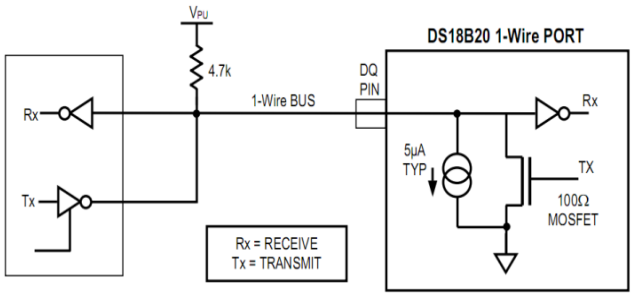
Podczas komunikacji z modułem ZigBee do tej ramki dodaje się jeszcze trzy dodatkowe pola. Pierwsze z nich, dołączane na samym początku ramki, determinuje rodzaj przesyłanej wiadomości (dla stosu ZigBee jest to *EZSP indicator* o wartości 0xFE), następne zawiera długość ramki podawaną w licznie transmitowanych bajtów. Na ostatniej pozycji ramka jest dopełniana wartością 0xA7, która oznacza koniec wiadomości. Po odpowiedniej konfiguracji wstępnej układu, jego zresetowaniu czy wyprowadzeniu ze stanu *sleep* oraz właściwym skonfigurowaniu sieci, przesyłana z komputera informacja może być umieszczana w polach parametrów $P_0 \div P_{N-1}$ i tak przesyłana dalej do modułu końcowego sieci za pośrednictwem komunikacji bezprzewodowej.

4. CZUJNIKI POMIAROWE

Do pomiaru temperatury posłużono się małym, cyfrowym czujnikiem DS18B20. Układ ten komunikuje się z hostem za pomocą interfejsu 1-wire. Do komunikacji służy jeden przewód sygnałowy, którym naprzemiennie przesyłane są do czujnika polecenia i odbierane od niego dane pomiarowe. Ponieważ wymiana informacji zrealizowana jest za pomocą tylko jednej linii sygnałowej podpiętej do zasilania przez rezystor 4,7 kΩ ,wymagana jest odpowiednia konfiguracja podłączonego do tej linii portu

I/O mikrokontrolera (rys.2) [2]. Wyjście typu otwarty dren umożliwia bezpieczne zwalnianie linii sygnałowej po przesłanym poleceniu i przejście w stan oczekiwania na odpowiedź czujnika.

5.



Rys. 2. Połączenie czujnika DS18B20 z mikrokontrolerem

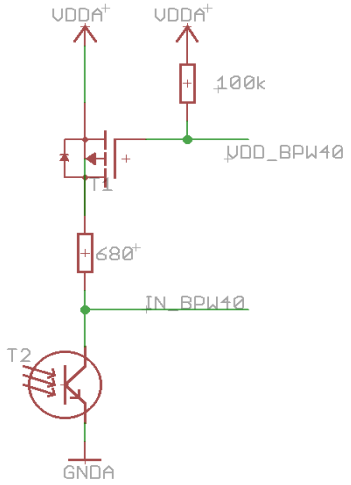
Układ DS18B20 zawiera aż trzy rodzaje pamięci. Pamięć ROM o rozmiarze 64 bitów zawiera Serial Code, na który składa jeden bajt informujący o rodzinie urządzenia oraz sześć bajtów zawierających jego numer seryjny, który pozwala na identyfikację czujnika w układach, w których do magistrali 1-wire podłączona jest większa liczba czujników. Ostatni bajt zawiera sumę CRC poprzednich siedmiu bajtów. Drugi rodzaj pamięci to pamięć nieulotna EEPROM, mogąca przechowywać zawartości trzech rejestrów konfiguracyjnych T_L, T_H i T_R nawet po zaniku zasilania układu. Rejestry T_L i T_H przechowują odpowiednio minimalną i maksymalną , odpowiednio, wartość temperatury, po przekroczeniu której ustawiana jest flaga alarmowa, którą host może w każdej chwili odczytać. Rejestr T_R określa rozdzielczość pomiaru temperatury. Możliwe rozdzielczości i odpowiadające im maksymalne czasy pomiaru temperatury przedstawiono w tab. 3.

Tablica 3. Zależność rozdzielczością od czasu pomiaru temperatury dla czujnika DS18B20

Rozdzielczość [bit]	9	10	11	12
Czas [ms]	93,75	187,5	375	750

Trzecim rodzajem pamięci jest pamięć ulotna RAM składająca się dziewięciu bajtów. Najistotniejsze dwa pierwsze bajty zawierają zakodowaną w postaci słowa bitowego wartość temperatury. Długość tego słowa zależy od wartości aktualnie przechowywanej w piątym z kolei bajcie tej pamięci (T_R), a którą można skonfigurować przesyłając ją bezpośrednio lub kopiując z pamięci EEPROM. Podobnie jest z bajtami o numerach 3 i 4, które przechowują dopuszczalne granice zakresu temperatury. Bajty o numerach 6,7 i 8 są zarezerwowane na potrzeby własne układu i nie jest możliwe ich zapisywanie. Ostatni, dziewiąty bajt zawiera sumę CRC obliczoną na podstawie poprzedzających go ośmiu bajtów.

Drugim wykorzystanym w systemie czujnikiem jest czujnik badający natężenie padającego na niego światła. Rolę czujnika pełni fototranzystor (T2) w układzie wspólnego emitera. Na podłączonym do kolektora rezystorze odkłada się napięcie proporcjonalne do przepływającego przez niego prądu, a ten z kolei zależy od natężenia padającego na fototranzystor światła (rys. 3). Niezwykle ważne jest odpowiednie dobranie wartości rezystora obciążającego. Zbyt duża jego wartość spowoduje przekroczenie maksymalnej wartości prądu przepływającego przez kolektor przy maksymalnym natężeniu światła. Z kolei zbyt mała wartość rezystancji będzie skutkowała nie wykorzystaniem całego dostępnego zakresu pomiarowego. Znając maksymalny prąd płynący przez kolektor fototranzystora oraz jego napięcie nasycenia, można wyznaczyć wymaganą wartość rezystancji.



Rys. 3. Schemat układu pomiarowego fototranzystorem BPW40

Podobnie jak dla czujnika DS18B20, fototranzystor zasilany jest napięciem 3.3V. W układach krytycznych pod względem zużycia energii zasilanych napięciem 3.3V należy zastosować odpowiedni klucz tranzystorowy odcinający w odpowiednich przedziałach czasu zasilanie od czujnika. Zastosowany w przykładzie tranzystor MOSFET typu *p* BSP171 zapewnia stabilne działanie układu, a decydują o tym jego dwie bardzo ważne cechy:

1. W stanie pełnego otwarcia rezystancja dren-źródło tranzystora wynosi zaledwie kilka mΩ, podczas gdy w stanie zatkania liczona jest w MΩ.
2. Napięcie wystarczające do pełnego otwarcia układu wynosi maksymalnie 2.0V.

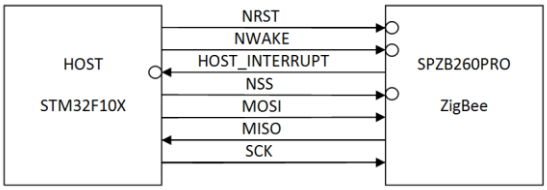
Sterowanie bramką tranzystora odbywa się przez ustawienie odpowiedniej wartości na porcie mikrokontrolera, a duża wartość rezystancji podłączonej do zasilania zapewni małe straty energii.

Sygnał pomiarowy z kolektora trafia bezpośrednio na wejście wbudowanego w mikrokontroler STM32F103RBT6 przetwornika ADC. Oczywiście możliwe jest zastosowanie przedwzmacniacza dla sygnału wyjściowego czujnika, jednakże bezpośredni pomiar sygnału zapewnia zupełnie wystarczającą dokładność. Ponieważ napięcie referencyjne przetwornika ADC jest równe napięciu zasilania całego modułu, wynikowe napięcie pomiaru natężenia światła mieści się w zakresie 0,2 ÷ 3,3 V. Znając zależność prądu kolektora od natężenia padającego światła można z dobrą dokładnością wyznaczyć wartość natężenia padającego na fototranzystor światła.

5. KOMUNIKACJA BEZPRZEWODOWA

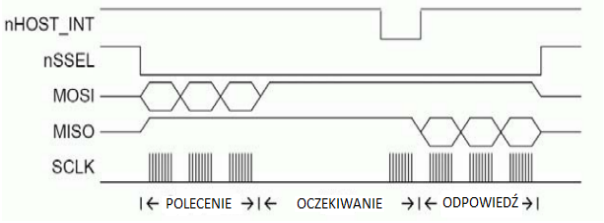
Moduł ZigBee połączony jest z hostem za pomocą siedmiu lini (rys. 4): NSS – linia wyboru urządzenia (aktywny poziom niski), SCK – sygnał zegarowy SPI, MISO – wejście urządzenia *master*, wyjście urządzenia *slave*, MOSI – wyjście urządzenia *master*, wejście urządzenia *slave*, NRST – linia resetująca moduł ZigBee (aktywny poziom niski), NWAKE – linia wybudzająca moduł ze stanu *sleep* (aktywny poziom niski), NHOST_INTERRUPT – przerwanie generowane przez moduł ZigBee.

Przebieg komunikacji można podzielić na trzy etapy (rys. 5) [3]. W pierwszym host wysyła polecenie do modułu ZigBee po wcześniejszym obniżeniu poziomu sygnału na lini NSS. W trakcie wysyłania kolejnych bajtów w odpowiedzi otrzymuje stałą wartość równą 0xFF. Po przesłaniu ostatniego bajtu procedura przechodzi do etapu oczekiwania. Jest to czasu, w którym moduł SN260 dekoduje otrzymane polecenie oraz wykonuje związaną z nim akcję.



Rysunek 4. Połącznie modułu ZigBee z mikrokontrolerem

Zakończenie tego etapu może zostać rozpoznane na dwa sposoby. Pierwszy polega na ciągłym odbiorze transmitowanych przez *slave* sygnałów i porównywaniu ich wartości z liczbą 0xFF, bowiem na tym etapie *slave* będzie przesyłał jedynie tę wartość. Pierwszy odebrany bajt różny od tej wartości będzie oznaczał przejście do kolejnego etapu. Drugim sposobem jest podsłuch linii HOST_INTERRUPT, bowiem gdy odpowiedź jest już gotowa moduł ZigBee obniża poziom sygnału na tej linii. W ostatnim etapie komunikacji, host odbiera kolejne bajty odpowiedzi a ich format jest zgodny z formatem protokołu EZSP. Ponieważ w tym czasie należy taktować linię zegarową SPI, zaleca się, aby host transmitował wartość 0xFF na linii MOSI i jednocześnie odbierał przychodzące na linii MISO kolejne bajty odpowiedzi. Po odbiorze bajtu kończącego wiadomość (czyli wartości 0xA7), host zwalnia linię NSS tym samym kończąc komunikację.



Rys. 5. Procedura komunikacji między modułem ZigBee i hostem

Znając procedurę komunikacji pomiędzy hostem i modułem ZigBee, konfiguracja sieci sprowadza się do wysłania kilku kolejnych poleceń zapoczątkowanych indykatorem EZSP i zawierających kolejne komendy konfiguracyjne oraz odbioru przychodzących wiadomości niosących często informację o statusie wykonania danej czynności. Przykładowo, procedura przyłączenia się do istniejącej sieci wygląda następująco:

sequence	=	0x00
frame control	=	0x00 (command frame)
joinNetwork command	=	0x1F
nodeType	=	0x02
extendedPanId	=	0x1122334455667788
panId	=	0x1234
radioTxPower	=	0xFF (-1 dB)
radioChannel	=	0x0B (11)

Host → SN260: | 00 | 00 | 1F | 02 | 88 | 77 | 66 | 55 |
| 44 | 33 | 22 | 11 | 34 | 12 | FF | 0B|

sequence	=	0x00
frame control	=	0x80 (response frame)
joinNetwork command	=	0x1F
status	=	0x00 (EMBER succes)

SN260 → Host: | 00 | 80 | 1F | 00 |

6. PODSUMOWANIE

Zaprezentowany system może posłużyć jako podstawa dużo bardziej rozbudowanych systemów. Funkcjonalność i prostota, z jaką EZSP umożliwia zarządzanie funkcjami stosu ZigBee sprawia, że konfiguracja sieci liczących wiele węzłów, jest możliwa nawet dla niezbyt zaawansowanych deweloperów tego typu systemów.

7. BIBLIOGRAFIA

1. Application Note 27: Understanding and Using Cyclic Redundancy Checks with Dallas Semiconductor iButtonTM Products, s.3; http://www.maxim-ic.com.
2. *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*, s.9 http://www.maxim-ic.com.
3. Paprocki K.: Mikrokontrolery STM32 w praktyce, Wydawnictwo BTC, Legionowo 2009, s. 175-180, ISBN 978-83-60233-52-8
4. EZSP Reference Guide, 17 October 2008, 120-3009-000G, www.ember.com.
5. RM0008 Reference manual, STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM - based 32-bit MCUs, www.st.com.

SENSOR CONTROLLING SYSTEM WITH APPLICATION WIRELESS NETWORK IN 802.15.4 STANDARD

Key-words: wireless sensor network, IEEE 802.15.4 standard, communication protocols

Summary: Wireless sensor networks are the most often used in measurement-control systems. A microcontroller with Cortex-CPU as a network controller and ZigBee transmitter working at 2.4 GHz have been applied in the presented system. Communication modules connecting the controller and the radio transmitter through SPI interface and a stack implementation related to requirements of the application had to been created. A software wirelessly controlling of automatics devices by means of the STM32 microcontroller in the IEEE 802.15.4 standard and the multifunctional controlling of the whole system from a PC have been prepared.