

## STEROWANIE CZUJNIKAMI W SIECI ZIGBEE PRZY POMOCY KOMEND GŁOSOWYCH

Jan ROGALSKI, Lech HASSE

Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki

Katedra Metrologii i Optoelektroniki

tel. +58 347 1884, fax: +58 341 6132

e-mail: lhasse@eti.pg.gda.pl

**Streszczenie:** Przedstawiono bezprzewodową sieć sensorową, pobierającą i przekazującą informacje z czujników temperatury i oświetlenia sterowanych głosem. Funkcjonalność systemu może zostać wykorzystana do sterowania urządzeniami automatyki domowej. System składa się z trzech zasadniczych elementów: koordynatora sieci ZigBee, urządzenia końcowego (pomiarowego) wyposażonego w czujniki światła i temperatury oraz komputera PC. Użytkownik może zarządzać systemem za pomocą trzech interfejsów: komend głosowych rejestrowanych przez mikrofon, wyświetlacza TFT z panelem dotykowym oraz komputera PC z opracowanym oprogramowaniem. W koordynatorze zastosowano uproszczony algorytm rozpoznawania izolowanych słów przy pomocy współczynników LPC (*Linear Predictive Coding*). System został skonstruowany na popularnych mikrokontrolerach firmy STM z nowoczesnym mikroprocesorem ARM Cortex-M3.

**Słowa kluczowe:** sensorowa sieć bezprzewodowa, standard IEEE 802.15.4, sterowanie głosem

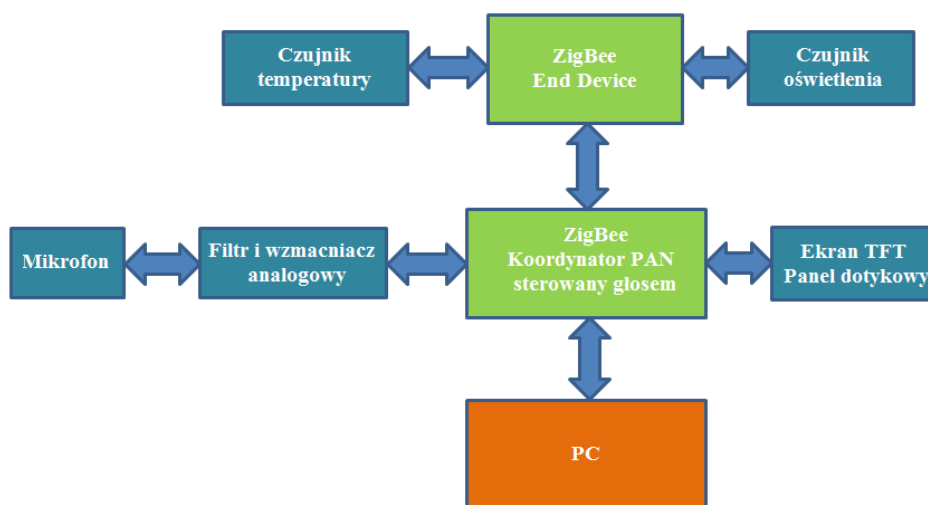
### 1. WPROWADZENIE

Systemy sterowania urządzeniami automatyki domowej od wielu lat zyskują na popularności. Ich zadaniem jest zwiększenie komfortu codziennego życia oraz minimalizowanie kosztów związanych z eksploatacją

domów i mieszkań. Wiele z komercyjnych rozwiązań wykorzystuje sieci bezprzewodowe do komunikacji pomiędzy elementami składającymi się na pełen system. Brak konieczności inwestycji w dodatkową infrastrukturę (np. okablowanie) w znacznym stopniu ułatwia implementację urządzeń w istniejących budynkach oraz obniża koszty instalacji. Niestety, komercyjne systemy automatyzacji posiadają ograniczenia wprowadzane przez producentów, którzy dopuszczają wykorzystywanie tylko elementów kompatybilnych z ich systemami. Powstała więc potrzeba stworzenia systemu pozwalającego na elastyczne zarządzanie urządzeniami oraz łatwą możliwość rozbudowy.

### 2. SYSTEM SIECI BEZPRZEWODOWEJ

Schemat blokowy działania systemu przedstawiono na rysunku 1. Składa się on z trzech zasadniczych elementów: koordynatora sieci ZigBee [1, 2], urządzenia końcowego (pomiarowego) wyposażonego w czujniki światła i temperatury oraz komputera PC.



Rys. 1. Ogólny schemat blokowy systemu

## 2.1. Koordynator sieci

ZigBee Koordynator sieci PAN (*Personal Area Network*) jest najważniejszą częścią całego systemu, tworząc sieć, do której podłączone są urządzenia końcowe oraz wykonując zadania ustalone przez użytkownika. Moduł został wykonany przy wykorzystaniu zestawu uruchomieniowego HY-MINI wyposażonego w mikrokontroler STM32F103VCT6, jeden z najmocniejszych w swojej rodzinie. Główną zaletą tego mikrokontrolera jest możliwość współpracy z 64 kB pamięcią RAM, co było niezbędne przy zaimplementowaniu w systemie algorytmu rozpoznawania mowy wymagającego przeprowadzania wielu operacji na dużej liczbie danych.

Koordynator posiada możliwość komunikacji zarówno z urządzeniem końcowym (przez sieć ZigBee) jak i z komputerem PC (przez interfejs USB), co oznacza, że pośredniczy on również w komunikacji pomiędzy komputerem a modułem pomiarowym. Za obsługę sieci ZigBee odpowiada również układ SPZB260PRO, natomiast za przesyłanie informacji w magistrali USB odpowiedzialny jest układ PL2302HX.

W celu ułatwienia obsługi interfejsu użytkownika oraz prezentacji wyników pomiarów, koordynator został wyposażony w 3,2-calowy wyświetlacz TFT z panelem dotykowym. Na ekranie wyświetlane jest menu, po którym można się poruszać za pomocą rysika. Z poziomu menu graficznego można skonfigurować pracę całego systemu oraz pobrać wyniki pomiarów z czujników zamontowanych na płytce modułu pomiarowego. Obsługa ekranu TFT jest realizowana przez dedykowany kontroler ILI9320 za pomocą interfejsu i80 opracowanego przez firmę Intel. Odczytywanie pozycji, w której nastąpiło „dotknięcie” ekranu również wymagało użycia zewnętrznego kontrolera. Do tego zadania zastosowano układ ADS7843, który wysyła wyznaczone parametry tej pozycji przez interfejs SPI (*Serial Protocol Interface*).

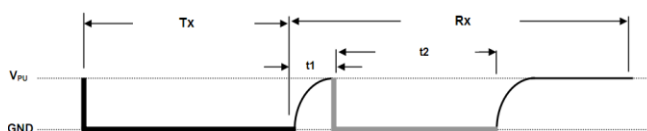
W wyjątkowych okolicznościach do poruszania się po menu układu można wykorzystać interfejs głosowy. Aby było to możliwe niezbędne jest podłączenie mikrofonu do układu wzmacniacza analogowego, którego zadaniem jest dopasowanie sygnału z mikrofonu do zakresu wartości napięcia przetwarzanych przez przetwornik A/C w mikrokontrolerze. Dodatkowo wzmacniacz filtruje sygnał eliminując szумы oraz tłumiąc składowe mocy sygnału leżące poza pasmem mowy, przyjętym jako pasmo użytkowe. Przy użyciu interfejsu głosowego można korzystać ze wszystkich opcji menu (analogicznie jak w przypadku użycia panelu dotykowego). Sterowanie odbywa się za pomocą 19 wcześniej zarejestrowanych komend zapisanych w pamięci RAM mikrokontrolera. Skuteczność rozpoznawania komend głosowych jest oczywiście największa, gdy użytkownik używający interfejsu głosowego wcześniej nagrał wzorce poleceń, które zostały zapisane do pamięci RAM.

## 2.2. Moduł pomiarowy

Sercem modułu pomiarowego sieci ZigBee (*END DEVICE*), jak i całego systemu, jest mikrokontroler STM32F103RBT6 (32-bitowy mikroprocesor ARM Cortex). Programowanie pamięci *flash* tego mikrokontrolera oraz debugowanie programu odbywa się przy użyciu 20-pinowego złącza JTAG, co pozwala na dostosowywanie programu do aktualnych wymagań użytkownika [3].

Do pomiaru natężenia światła został wykorzystany układ analogowy zawierający fototranzystor BPW40, podłączony bezpośrednio do przetwornika A/C w mikrokontrolerze.

Za pomiar temperatury odpowiedzialny jest cyfrowy czujnik DS18B20s. Ten popularny układ używa interfejsu 1-wire do komunikacji z mikrokontrolerem. Producent umożliwił konfigurację kilku parametrów pracy czujnika, takich jak m. in. rozdzielczość pomiaru czy też maksymalna oraz minimalna temperatura, przy której nie występuje alarm informujący o przekroczeniu wybranego zakresu. Przykładowo, sterowanie czujnika temperatury DS18B20 rozpoczyna się od inicjalizacji (rys. 2), w przedziale czasu



Rys. 2. Przebieg czasowy inicjalizacji czujnika temperatury DS18B20

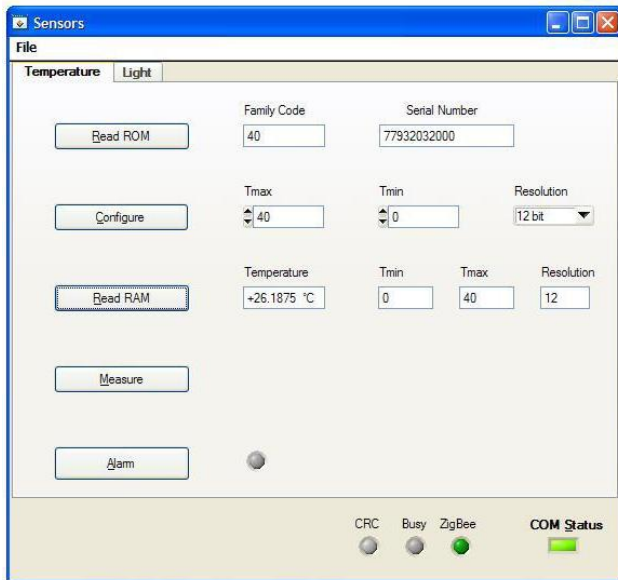
Tx master (host) obniża poziom sygnału na linii 1-Wire na co najmniej 480  $\mu$ s, co jest rozumiane przez czujnik jako polecenie resetu. Po upływie czasu Tx master zwalnia magistralę, a podłączony do niej rezystor *pull-up* powoduje wzrost potencjału aż do wartości napięcia zasilania (3.3 V). W przypadku prawidłowej operacji resetu, czujnik odpowiada po upływie czasu t<sub>1</sub> (15÷60  $\mu$ s) obniżając poziom sygnału na magistrali na okres czasu 60÷240  $\mu$ s. Również procedura odczytu danych z magistrali 1-Wire jest inicjowana przez urządzenie master. Po uprzednim przesłaniu odpowiedniej komendy, obniżenie poziomu sygnału na magistrali jest dla czujnika znakiem, że może on rozpocząć transmisję pierwszego bitu danych. Master zwalnia magistralę już po upływie 2  $\mu$ s, a próbkowanie następuje po około 13  $\mu$ s.

Komunikacja bezprzewodowa realizowana jest za pomocą zewnętrznego układu SPZB260PRO firmy ST, zawierającego procesor sieciowy zgodny ze standardem IEEE 802.15.4 (*ZigBee*) oraz antenę zdolną do transmisji danych z mocą nominalną 3 dBm oraz czułością na poziomie -95 dBm (co pozwala na przesyłanie wiadomości na dystansie kilkudziesięciu metrów). Komunikacja z mikrokontrolerem odbywa się przy użyciu interfejsu SPI przez protokół EZSP (*Ember ZNet Serial Protocol*), który umożliwia wykorzystanie zaimplementowanego w procesorze stosu ZigBee. Protokół ten pozwala na konfigurację wielu parametrów niezbędnych do prawidłowego funkcjonowania całej sieci.

Urządzenie pomiarowe zostało wyposażone w układ zasilania składający się z czterech baterii typu AA 1.5 V.

## 2.3. Funkcje komputera PC

Komputer PC wraz z opracowaną aplikacją desktopową służy do konfiguracji pracy czujników (ustalania wymaganego zakresu parametrów) oraz do odczytywania wyników pomiarów. Opracowany interfejs użytkownika pozwala na intuicyjne sterowanie pracą całego systemu oraz obsługę wirtualnego portu szeregowego (*Virtual Port COM*) umożliwiającego zrealizowanie komunikacji z koordynatorem za pomocą interfejsu USB. Oprogramowanie zostało wykonane w środowisku LabVIEW w formie panelu przedstawionego na rys. 3.



Rys. 3. Okno aplikacji desktopowej

### 3. ALGORYTM ROZPOZNAWANIA KOMEND GŁOSOWYCH

Najtrudniejszym zadaniem podczas realizacji projektu było opracowanie i implementacja metody rozpoznawania komend głosowych wydawanych przez użytkownika systemu [4, 5]. Głównym ograniczeniem, z jakim należało się zmierzyć, był brak wsparcia dla operacji DSP w mikrokontrolerze firmy STM zastosowanym w kordynatorze. W związku z ograniczonymi zasobami sprzętowymi należało wybrać metodę przetwarzania, która nie wymagałaby dużej liczby złożonych obliczeń, byłaby odporna na ewentualne błędy wynikające z zaokrąglania wyników obliczeń, nie wymagałaby stosowania bardzo czasochłonnej operacji dzielenia liczb, zużywałaby relatywnie mało pamięci RAM, a przy tym była skuteczna.

Większość komercyjnych systemów rozpoznawania mowy realizowana jest na komputerach desktopowych posiadających duże moce obliczeniowe lub na dedykowanych procesorach DSP. Najlepsze efekty uzyskuje się przy wykorzystaniu algorytmów opartych o teorię *sieci neuronowych* lub *ukrytych modeli Markova*. Niestety, obydwa te rozwiązania byłyby bardzo trudne do implementacji na mikrokontrolerze ze względu na dużą złożoność obliczeniową tych algorytmów. W kordynatorze został ostatecznie zastosowany uproszczony algorytm rozpoznawania izolowanych słów oparty o metodę kodowania sygnałów mowy za pomocą współczynników LPC. Wybrany algorytm składa się z kilku etapów.

Pracę z systemem rozpoczyna się od nagrania wszystkich komend niezbędnych do poprawnego poruszania się po menu systemu za pomocą mikrofonu. Tworzy się w ten sposób książkę kodową (*Code Book*) zawierającą wzorce wykorzystywane w kolejnych krokach algorytmu. Składa się ona z 19 różnych komend zapisanych w pamięci RAM mikrokontrolera. Polecenia zajmują w sumie 24,5 kB, czyli ponad połowę dostępnej pamięci. Maksymalny czas trwania pojedynczego dźwięku wynosi 0,7s.

Każde nagrywane słowo jest próbkowane z częstotliwością 8000 Hz oraz poddawane preemfazie (analogicznie jak w filtrach psfometrycznych) w celu uwydatnienia wyższych częstotliwości, które mają mniejszą

amplitudę. Uzyskuje się dzięki temu przeskalowanie mocy sygnału mowy ustalając podobny poziom odniesienia dla częstotliwości z przenoszonego pasma. Operacja ta, ponadto, pozwala na redukcję poziomu szumów, które występują na wyjściu przetwornika A/C. Preemfaza jest dokonywana za pomocą nierekursywnego cyfrowego filtra FIR o transmittancji:

$$s(n) = s(n) - 0,9375s(n-1) \quad (1)$$

gdzie:  $n$  – numer badanej próbki,  $s(n)$  – wartość badanej próbki

Następnie zarejestrowany sygnał dzielony jest na 33 ramki (odpowiada to okienkowaniu za pomocą okna prostokątnego). Niestety, w wyniku takiego okienkowania zachodzi duże zniekształcenie sygnału w dziedzinie czasu, a co za tym idzie, przeciek widma w dziedzinie częstotliwości. Główną przyczyną negatywnego wpływu okna prostokątnego na analizowany fragment mowy jest brak ciągłości sygnału na krańcach ramki. Aby pozbyć się tego niekorzystnego efektu zastosowano mnożenie sygnału z oknem Hamminga o współczynnikach:

$$w(n) = 0.53836 - 0.46164 \cdot \cos\left(\frac{2\pi n}{N-1}\right) \quad (2)$$

gdzie  $n$  – numer współczynnika,  $N$  – liczba próbek. Pozwala to zachować ciągłość analizowanego sygnału na jego krańcach i dzięki temu znacznie ogranicza przeciek, który wystąpił podczas fragmentowania dźwięku mowy.

Dla sygnału odfiltrowanego wyznacza się funkcję autokorelacji:

$$r(k) = \sum_{n=0}^{N-1-k} s(n) * s(n+k) \quad k = 0, 1, 2, \dots, 10 \quad (3)$$

gdzie, podobnie jak poprzednio,  $n$  – numer aktualnej próbki,  $N$  – liczba próbek.

Zwykle wykorzystuje się to narzędzie do wykrycia sygnału okresowego w mocno zaszumionej próbce dźwięku lub do wyznaczenia dominującej częstotliwości w sygnale. W przypadku kodowania LPC wykorzystywanych jest pierwszych dziesięć współczynników funkcji autokorelacji uzyskiwanych z każdej ramki do wyznaczenia niezbędnych parametrów filtra traktu głosowego.

Ostatnim etapem algorytmu jest wyznaczenie dziesięciu współczynników LPC charakteryzujących daną ramkę. Klasyczna metoda wymaga rozwiązanie skomplikowanego równania macierzowego. W celu uproszczenia algorytmu rozpoznawania mowy, w kordynatorze zaimplementowano iteracyjną metodę *Durbina-Levisona* [6] w celu wyznaczenia tych współczynników. Pozwala ona na zwiększenie skuteczności oraz czasu rozpoznawania poleceń. Po przeprowadzeniu operacji dla wszystkich ramek nagranych polecenia otrzymuje się 330 współczynników.

Najprostszym sposobem by rozpoznać wypowiedziane słowo jest porównanie go z wcześniej zarejestrowanymi wzorcami i znalezienie takiego, który najlepiej pasuje do badanego słowa. Niestety, wiele czynników ma wpływ na to, czy uda się poprawnie przeprowadzić operację rozpoznawania. Po pierwsze, za każdym razem, gdy wypowiedzane jest dane słowo, trwa to nieco krócej bądź dłużej niż wzorec. Można wyeliminować ten problem przez ustalenie czasu nagrywania danego słowa. Dzięki temu

wszystkie próbki będą posiadać tę samą długość, co upraszcza algorytm porównywania. Aby rozwiązać ten problem, w koordynatorze wykorzystano algorytm DTW (*Dynamic Time Warping*) polegający na nieliniowym porównywaniu fragmentów nagranych dźwięku ze wzorcami. Dla każdego porównania otrzymywana jest liczba oznaczająca podobieństwo między wzorcem a badanym dźwiękiem.

#### 4. PROGRAM SCHEDULER

W systemie wykorzystywanych jest wiele układów peryferyjnych, takich jak wyświetlacz, panel dotykowy, diody LED, itp. Każdy z tych peryferii musi zostać obsłużony w odpowiednim czasie. W celu uproszczenia czasowej obsługi tych urządzeń zaimplementowano proste środowisko do kontrolowania zadań wykonywanych przez mikrokontroler – *Scheduler*.

Jest to program pełniący podobne zadanie w systemie mikroprocesorowym co system operacyjny w komputerze PC. Z niskopoziomowego punktu widzenia, można go określić, jako program obsługi przerw ( *Interup Service Routine* ) przychodzących od jednego z układów *timera* mikrokontrolera, wykorzystywany przez wszystkie inne funkcje systemu, co pozwala na jednorazowe lub okresowe wykonywanie dedykowanych zadań. Gdy zadanie jest gotowe do wykonania, wstawiane zostaje na tzw. listę oczekującą. Znajdują się na niej funkcje czekające aż nadjedzie ich kolej do skorzystania z zasobów mikrokontrolera. W dobrze dopasowanym systemie, zadanie jest wykonywane natychmiast bez żadnej zwłoki. Po zakończeniu pracy, oddaje ono kontrolę *Schedulerowi*. Ważne jest, aby zwrócić uwagę na brak wyłączenia zadań, co oznacza, że każda funkcja zostanie wykonana bez przerywania.

Zaimplementowany w systemie *Scheduler* zawiera trzy główne funkcje obsługujące wszystkie zdarzenia zachodzące w systemie:

- *int Scheduler\_Add\_Task* – dodaje zadanie do listy procesów wykonywanych przez mikrokontroler (np. obsługa wyświetlacza, RS232 itp.); to właśnie na etapie dodawania zadań zachodzi planowanie kolejności ich wykonywania,

- *void Scheduler\_Uptdate\_Task\_List* – obsługuje przerwania przychodzące od timera; zmniejsza ona wartość liczników informujących o czasie do następnego wykonania każdego z procesów,
- *void Scheduler\_Dispatch\_Tasks* – sprawdza listę procesów w poszukiwaniu tych, które są gotowe do wykonania.

Zanim zainicjowana zostanie praca z wykorzystaniem *Schedulera*, niezbędne jest odpowiednie ustawienie *timera* sterującego jego pracą

#### 5. WNIOSKI KOŃCOWE

Przedstawiony w pracy prototyp prostego systemu sterowania głosem urządzeniami automatyki domowej stanowi alternatywę dla komercyjnych rozwiązań znajdujących się na rynku. Jego główną zaletą jest elastyczne zarządzanie urządzeniami wykonawczymi oraz łatwa możliwość rozbudowy systemu o dodatkowe elementy.

#### 5. BIBLIOGRAFIA

1. Karl H., Andreas Willing: *Protocols and architectures for wireless sensor networks*. Wiley 2005, 526 s., ISBN: 0-470-09510-5.
2. Gislason D.: *ZigBee wireless networking*. Newness Press, 2008, 448 s., ISBN 13: 978-0-7506-8597-9.
3. Parol B., Hasse L.: *System sterowania czujnikami z wykorzystaniem sieci bezprzewodowej w standardzie IEEE 802.15*, Zeszyty Naukowe Wydziału EiA PG, 2010, Nr 28, s. 119-122, ISSN 1425-5766.
4. Łączkowska M.: *Wprowadzenia do anatomii, fizjologii i patologii narządu głosu, mowy i słuchu*. Uniwersytet M. Curie-Skłodowskiej, Zakład Logopedii, Lublin 1986.
5. Keshet J., Bengio S. ed.: *Automatic speech and speaker recognition: Large margin and kernel methods*. Wiley 2009, 253 s., ISBN: 978-0-470-69683-5.
6. Delsarte, P. and Genin, Y. V.: *The split Levinson algorithm*, *IEEE Trans. on Acoustics, Speech, and Signal Processing*, v. ASSP-34(3), 1986, s. 470-478.

## CONTROLLING OF SENSORS IN THE ZIGBEE NETWORK BY MEANS OF VOICE COMMANDS

**Key-words:** wireless sensor network, IEEE 802.15.4 standard, voice controlling

**Summary:** Wireless sensor network collecting and transferring information from temperature and lighting sensors controlled by means of operator's voice has been presented. Functionality of the system can be applied for controlling of home automatic devices. The system consists of three main parts: ZigBee network coordinator, measuring terminals equipped in light and temperature sensors and computer PC. A user can manage the system by means of three interfaces: voice commands registered by the microphone, TFT display with touch panel and computer PC with elaborated software. In the coordinator the simplified algorithm of isolated words recognition by means of the LPC (*Linear Predictive Coding*) coefficients was applied. The system was realized using popular microcontrollers produced by STM with the modern microprocessor ARM Cortex-M3.