

**Studia i Materiały  
Informatyki Stosowanej**



# **Studia i Materiały Informatyki Stosowanej**

Czasopismo młodych pracowników  
naukowych, doktorantów i studentów

Tom 11, Nr 2, 2019

Bydgoszcz 2019

**Studia i Materiały Informatyki Stosowanej**  
**Czasopismo młodych pracowników naukowych, doktorantów**  
**i studentów**

© Copyright 2019 by Uniwersytet Kazimierza Wielkiego

**Patronat naukowy:**

Instytut Informatyki  
Uniwersytet Kazimierza Wielkiego  
ul. Chodkiewicza 30  
85-064 Bydgoszcz  
tel. (052) 325 76 11  
e-mail: [simis@ukw.edu.pl](mailto:simis@ukw.edu.pl)

**ISSN 1689-6300**

**Projekt okładki: Łukasz Zawadzki (StudioStrzelec.pl)**  
**DTP: Dawid Ewald**

**Wydawca:**

Instytut Informatyki  
Uniwersytet Kazimierza Wielkiego  
Dyrektor:  
dr hab. inż. Izabela Rojek, prof. uczelni  
ul. Chodkiewicza 30  
85-064 Bydgoszcz  
tel. +48 52 325 76 11  
*email: [izarojek@ukw.edu.pl](mailto:izarojek@ukw.edu.pl)*

**Kontakt:**

dr inż. Jacek Czerniak, prof. uczelni  
dr hab. inż. Marek Macko, prof. uczelni  
Uniwersytet Kazimierza Wielkiego  
ul. Chodkiewicza 30  
85-064 Bydgoszcz  
*e-mail: [jczerniak@ukw.edu.pl](mailto:jczerniak@ukw.edu.pl)*  
*[mackomar@ukw.edu.pl](mailto:mackomar@ukw.edu.pl)*

**Druk (ze środków sponsora):**  
**Oficyna Wydawnicza MW**

**Nakład 250 egz.**

**Bydgoszcz 2019**

**Studies and Materials  
in  
Applied Computer  
Science**

Journal of young researchers,  
PhD students and students

Vol. 11, No.2, 2019

Bydgoszcz 2019

**Studies and Materials in Applied Computer Science**  
**Journal of young researchers, PhD students and students**

© Copyright 2019 by Kazimierz Wielki University in Bydgoszcz

**Scientific patronage:**  
Institute of Informatics  
Kazimierz Wielki University  
ul. Chodkiewicza 30  
85-064 Bydgoszcz, Poland  
tel. +48 52 325 76 11  
*e-mail: simis@ukw.edu.pl*

**ISSN 1689-6300**

**Cover designed by: Łukasz Zawadzki (StudioStrzelec.pl)**  
**DTP by: Dawid Ewald**

**Publisher:**

Institute of Informatics  
Kazimierz Wielki University  
Head:  
Izabela Rojek, PhD DSc Eng Assoc. Prof.  
ul. Chodkiewicza 30  
85-064 Bydgoszcz, Poland  
tel. + 48 52 325 76 11  
*e-mail: izarojek@ukw.edu.pl*

**Contact:**

Jacek Czerniak, PhD. Eng. Assoc. Prof.  
Marek Macko, PhD. DSc. Eng., Assoc Prof  
Kazimierz Wielki University  
ul. Chodkiewicza 30  
85-064 Bydgoszcz, Poland  
*e-mail: jczerniak@ukw.edu.pl*  
*mackomar@ukw.edu.pl*

**Printing (funded from non-profit programme):**  
**Oficyna Wydawnicza MW**

**Edition of 250 copies**

**Bydgoszcz 2019**

# Studia i Materiały Informatyki Stosowanej

czasopismo młodych pracowników naukowych, doktorantów i studentów

patronat: Polskie Towarzystwo Informatyczne



## Przewodniczący Rady Naukowej

prof. dr hab. inż. czł. rzec. PAN Janusz Aleksander Kacprzyk, IBS PAN

## Redaktorzy Naczelni

dr inż. Jacek Czerniak, UKW  
dr hab. inż. Marek Macko, prof. nadzw.

## Redaktor Zarządzający

dr inż. Łukasz Apiecionek, UKW

## Redaktor Statystyczny

dr Iwona Filipowicz, UKW

## Komitet Redakcyjny

dr inż. Mariusz Dramski, AM  
dr inż. Hubert Zarzycki, WWSIS  
dr inż. Marcin Łukasiewicz, UTP  
dr inż. Piotr Dziurzański, ZUT  
dr inż. Tomasz Kałaczyński, UTP  
dr hab. inż. Grzegorz Domek, prof. nadzw.  
dr Piotr Prokopowicz, UKW

## Redaktor Tematyczny (Informatyka)

prof. dr inż. Rafał A. Angryk, GSU

## Redaktor Tematyczny (Mechatronika)

prof. dr.h.c.mult. Peter Kopacek, VUT

## Redaktor Tematyczny (Metody numeryczne)

dr hab. Petro Filevych, LNUVB

## Redaktor Językowy (j.polski)

dr Małgorzata Kempieńska, FRM

## Redaktor Językowy (j.angielski)

Andrew Gill, Reed Elsevier, UK

## Rada Naukowa

dr hab. Stanisław	Ambroszkiewicz	Instytut Podstaw Informatyki PAN
prof. dr inż. Rafał A.	Angryk	Georgia State University, USA
dr hab. Zenon	Biniek	Wyższa Szkoła Technologii Informatycznych
prof. dr hab. inż. Ryszard	Budziński	Zachodniopomorski Uniwersytet Technologiczny
dr inż. Joanna	Chimiak-Opoka	University of Innsbruck, Austria
prof. dr hab. inż. Ryszard	Choraś	Uniwersytet Technologiczno-Przyrodniczy
dr hab. Petro	Filevych	Lviv National University of Veterinary and Biotechnologies, Ukraina
prof. dr hab. inż. Piotr	Gajewski	Wojskowa Akademia Techniczna
dr inż. Marek	Holyński	Prezes Polskiego Towarzystwa Informatycznego
prof. dr hab. inż. czł. rzec. PAN Janusz	Kacprzyk	Instytut Badań Systemowych PAN
dr hab. inż. Andrzej	Kobyliński	Szkoła Główna Handlowa
prof. dr.h.c.mult. Peter	Kopacek	Vienna University of Technology, Austria
prof. dr hab. inż. czł. koresp. PAN Józef	Korbicz	Uniwersytet Zielonogórski
prof. dr hab. inż. Jacek	Koronacki	Instytut Podstaw Informatyki PAN
prof. dr hab. inż. Marek	Kurzyński	Politechnika Wrocławska
prof. dr hab. inż. Halina	Kwaśnicka	Politechnika Wrocławska
prof. dr Miroslaw	Majewski	New York Institute of Technology, United Arab Emirates
dr inż. Dariusz	Mikolajewski	Uniwersytet Kazimierza Wielkiego
prof. dr hab. Andrzej	Marciniak	Politechnika Poznańska
dr hab. Marcin	Paprzycki	Instytut Badań Systemowych PAN
prof. dr hab. inż. czł. PAN Witold	Pedrycz	University of Alberta, Canada
prof. dr hab. inż. Andrzej	Piegat	Zachodniopomorski Uniwersytet Technologiczny
prof. dr hab. inż. Andrzej	Polański	Politechnika Śląska
prof. dr hab. inż. Orest	Popov	Zachodniopomorski Uniwersytet Technologiczny
prof. dr inż. George	Przybył Einstein	College of Medicine, USAT Montserrat
dr hab. inż. prof. nadzw. Izabela	Rojek	Uniwersytet Kazimierza Wielkiego
prof. dr hab. inż. Danuta	Rutkowska	Politechnika Częstochowska
prof. dr hab. inż. czł. koresp. PAN Leszek	Rutkowski	Politechnika Częstochowska
prof. dr hab. inż. Milan	Sága	Žilinská Univerzita, Słowacja
prof. dr hab. inż. czł. rzec. PAN Roman	Słowiński	Instytut Badań Systemowych PAN, Politechnika Poznańska
prof. dr hab. inż. Włodzimierz	Sosnowski	Uniwersytet Kazimierza Wielkiego, IPPT PAN
prof. dr hab. inż. Andrzej	Stateczny	Akademia Morska w Szczecinie
dr hab. inż. Jan	Studziński	Instytut Badań Systemowych PAN
prof. dr hab. Tomasz	Szapiro	Szkoła Główna Handlowa
dr hab. Janusz	Szczepański	Uniwersytet Kazimierza Wielkiego, IPPT PAN
prof. dr hab. inż. czł. rzec. PAN Ryszard	Tadeusiewicz	Akademia Górniczo-Hutnicza
prof. zw. dr hab. inż. czł. rzec. PAN Jan	Węglarz	Instytut Chemii Bioorganicznej PAN, Politechnika Poznańska
prof. dr hab. inż. Sławomir	Wierchoń	Instytut Podstaw Informatyki PAN
prof. dr hab. inż. Antoni	Wiliński	Zachodniopomorski Uniwersytet Technologiczny
dr hab. inż. Andrzej	Wiśniewski	Uniwersytet Przyrodniczo-Humanistyczny w Siedlcach
dr hab. inż. Ryszard	Wojtyna	Uniwersytet Technologiczno-Przyrodniczy
prof. dr hab. Sławomir	Zadrożny	Instytut Badań Systemowych PAN
prof. dr. inż. Milan	Žmindák	Žilinská Univerzita, Słowacja
prof. dr hab. Zenon	Zwierzewicz	Akademia Morska w Szczecinie





## SPIS TREŚCI

<b>Od Redakcji</b> .....	4
<b>Programowalna platforma mobilna do nauki programowania</b> Daniel Chudziński, Piotr Kotlarz, Marcin Kempieński .....	5
<b>Projekt systemu monitorowania obturacyjnego bezdechu sennego</b> Przemysław Szyszkowski, Grzegorz Śmigielski .....	10
<b>Rozproszone systemy baz danych w planowaniu produkcji</b> Izabela Rojek, Zbyszko Królikowski, Mariola Marciniak .....	14
<b>Grupy jednolite w metodzie reprezentacji binarnych</b> Janusz Łukowski .....	21



## OD REDAKCJI

*Szanowni Czytelnicy,*

*Przedstawiamy Czytelnikom kolejny numer naszego czasopisma poświęcony szerokiemu przekrojowi zagadnień z obszaru informatyki: od reprezentacji binarnych, poprzez zagadnienia programistyczne (w tym nauki programowania) aż do przemysłowego wykorzystania rozproszonych systemów baz danych. Poszerza to perspektywę spojrzenia na zagadnienia z obszaru informatyki jako rozwiązywania szerokiego spektrum problemów: od teoretycznych aż po złożone kwestie praktyczne generowane przez przedsiębiorstwa i instytucje z otoczenia społecznego i gospodarcze uczelni. Pozwala to głębiej spojrzeć na potencjał rozwojowy informatyki jako nauki stosowanej, przyczyniającej się do poprawy kondycji gospodarczej miasta, regionu i kraju, w w szerszej perspektywie - globalnej jakości życia ludzi.*

Redaktorzy Naczelni SiMIS,  
dr inż. Jacek Czerniak,  
dr hab. inż. Marek Macko, prof. nadzw.

# PROGRAMOWALNA PLATFORMA MOBILNA DO NAUKI PROGRAMOWANIA

Daniel Chudziński<sup>\*1</sup>, Piotr Kotlarz<sup>2</sup>, Marcin Kempński<sup>2</sup>

<sup>1</sup> (Student)Uniwersytet Kazimierza Wielkiego, Instytut Mechaniki i Informatyki Stosowanej, Chodkiewicza 30, 85-064 Bydgoszcz

<sup>2</sup> Uniwersytet Kazimierza Wielkiego, Instytut Mechaniki i Informatyki Stosowanej, Chodkiewicza 30, 85-064 Bydgoszcz  
e-mail: piotrk@ukw.edu.pl

**Streszczenie:** W artykule przedstawiono możliwości wykorzystania niskobudżetowych technologii do budowy programowalnych kołowych platform mobilnych, między innymi do zastosowań edukacyjnych. Prototyp opracowanej platformy porusza się z wykorzystaniem dwóch silników DC, komputer pokładowy to moduł Arduino Uno.

**Słowa kluczowe:** Arduino, programowanie, uczenie się

## *Programmable mobile platform for learning programming*

**Abstract:** The article shows possibilities of application of low-cost technologies to construction programmable wheeled mobile platforms for multipurpose use, including learning. The prototype moves thanks to use of two DC motors. Onboard computer is based on Arduino Uno.

**Key words:** Arduino, programming, learning

## 1. Wprowadzenie

W ramach tej pracy przeprowadzamy analizę możliwości zastosowania niskobudżetowych układu elektroniki do budowy autonomicznych, programowalnych platform mobilnych. Opracowany został projekt oraz prototyp platformy mobilnej, która jest kontrolowana poprzez autorskie oprogramowanie dla systemu Android. Opracowano implementację protokołów wymiany danych w zakresie sterowania platformą w czasie rzeczywistym, oraz dla potrzeb akwizycji danych. W zakresie rozwiązań sprzętowych opracowano układ sterowania, zasilania oraz zestaw czujników dostosowanych do programowania ruchu platformy oraz do jej autonomicznego poruszania się. Założeniem było opracowanie prototypu platformy zarówno w wymiarze sprzętowym i programistycznym możliwej do zastosowania w edukacji na poziomie szkoły średniej oraz studiów kierunków inżynierskich. Tematyka zastosowań programowalnych platform mobilnych jest obecnie bardzo popularna w światowych pracach B+R i badaniach naukowych [1][2].

## 2. Zastosowane technologie i sytuacja licencyjna

W zakresie modułu sprzętowego wybrano Platformę Arduino [3] ze względu na dużą dostępność na rynku zarówno mikrokontrolerów jak i akcesoriów.

Do programowania urządzenia użyto programu dedykowanego przez Arduino i dostępnego na stronie [www.arduino.cc](http://www.arduino.cc). Zarówno platforma Arduino [4] jak i kompilator to popularne rozwiązania, które pozwalają zautomatyzować pewne czynności z zakresu prostej robotyki i automatyki bez znajomości zagadnień sterowania i mechaniki, oraz przy minimalnej wiedzy z zakresu elektroniki.

Aplikacja do kontroli platformy mobilnej została napisana dla systemu Android, który cechuje się dużą popularnością w rozwiązaniach mobilnych. Do napisania aplikacji użyto programu AndroidStudio [5], który jest dedykowanym środowiskiem deweloperskim dostarczonym przez firmę Google.

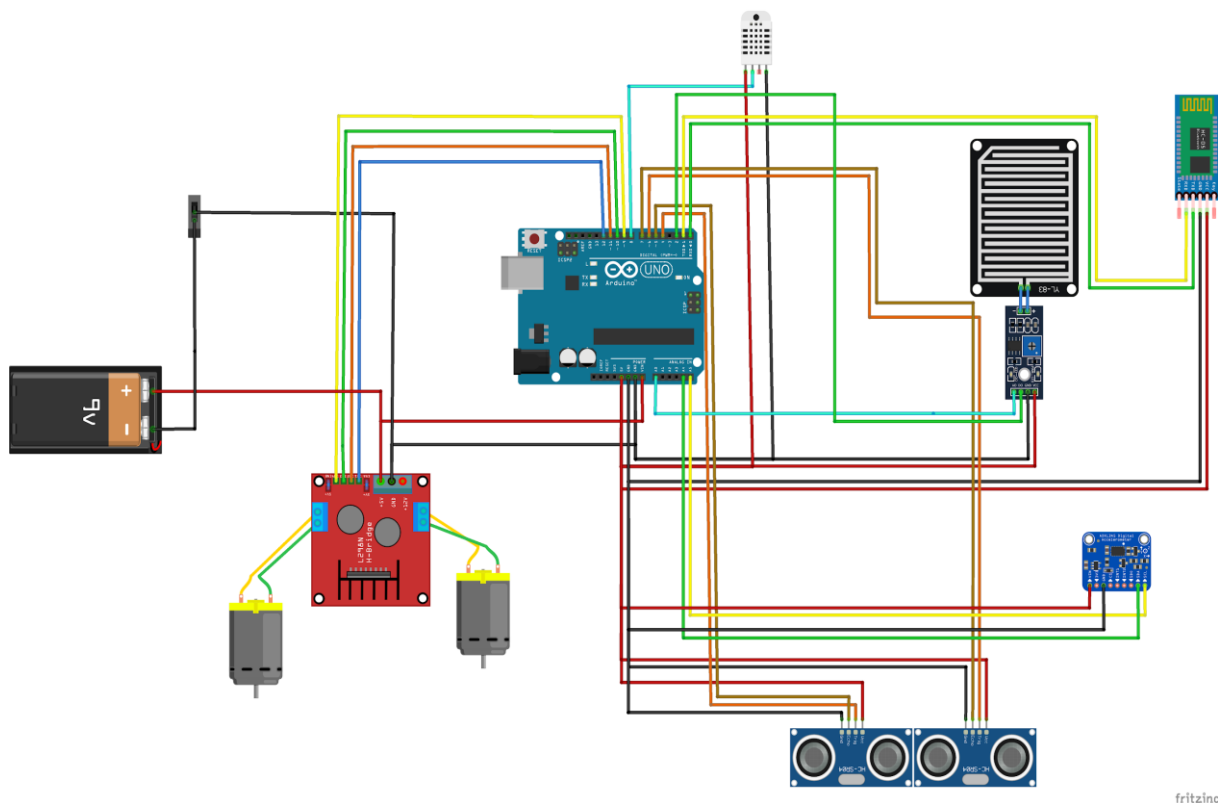
Jako silnik bazy danych zastosowano SQLite jest to system zarządzania bazą danych, który obsługuje język SQL dostępny w ramach licencji public domain. SQLite posiada API między innymi dla języka programowania Java, który jest wykorzystywany w tym projekcie. Baza znajduje się w pliku lokalnie zapisanym na urządzeniu.

Kwestie licencyjne, które występują w projekcie przedstawiają się następująco: SQLite działa na podstawie licencji public domain, natomiast Android Studio działa na licencji Freeware. Jeżeli chodzi o platformę Arduino to działa ono na licencji GNU Lesser General Public License (LGPL) tak samo jak sam język programowania Java. W programie Arduino

wykorzystywane są biblioteki ADXL345.h (odpowiada za obsługę akcelerometru) oraz DHT.h (odpowiada za odczyt temperatury i wilgotności z czujnika DHT22), które są udostępniane użytkownikom na podstawie licencji MIT. Natomiast w aplikacji Android zastosowana została biblioteka Gauge, która odpowiada za wizualizację pomiarów i jest dostępna w licencji Apache.

### 3. Moduł sprzętowy projektu

Sercem moduły sprzętowego jest układ komputera jednopłytkowego Arduino Uno, poniżej przedstawiono i opisano kluczowe kwestie realizacji całego moduły sprzętowego.



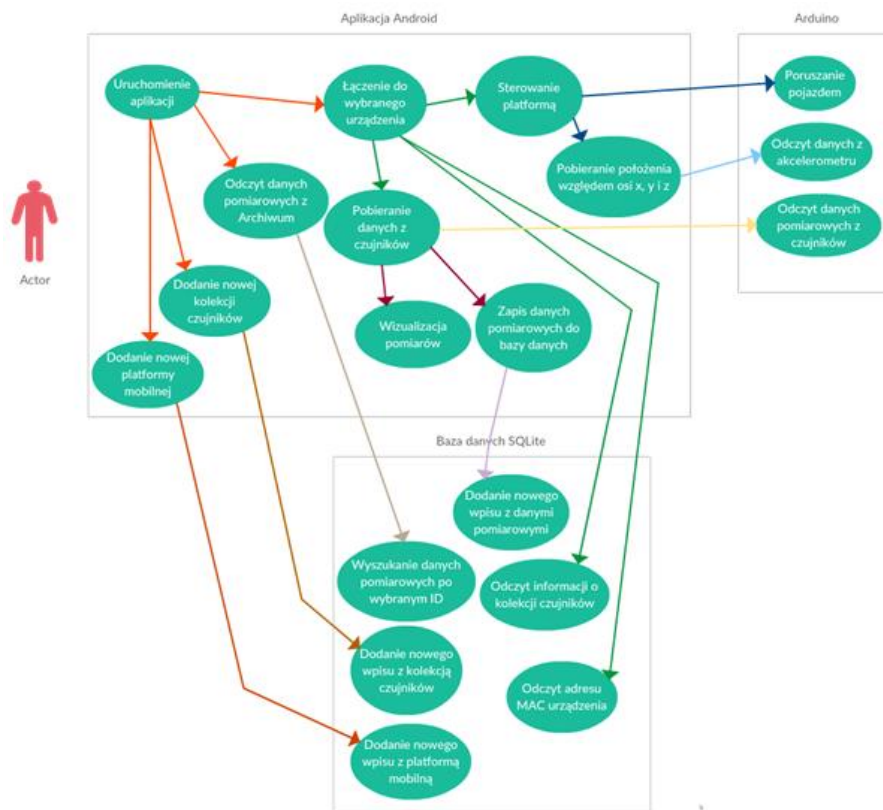
**Rysunek 1.** Schemat połączeniowy układu sterowania i akwizycji danych

Sterownik silników L298N jest podłączony bezpośrednio do układu zasilania natomiast piny wejściowe sterownika podłączone są do Arduino Uno, a dokładniej do pinów cyfrowych o numerach 9, 10, 11 oraz 12. Do sterownika są podłączone dwa silniki DC, które będą poruszać się do przodu lub do tyłu w zależności od tego, na który port wejściowy sterownika ustawimy stan wysoki. Samo Arduino jest także zasilane zestawem baterii 4xAAA i regulowane włącznikiem. Czujnik temperatury i wilgotności DHT22 jest podłączony do portu cyfrowego o numerze 8 znajdującego się w mikrokontrolerze, a zasilany jest napięciem 5V. Czujnik deszczu, śniegu i wody FC-37 podłączony jest zarówno do portu analogowego A0 oraz do portu cyfrowego o numerze 2, a zasilany jest także napięciem 5V udostępnianego z Arduino. Pierwszy czujnik odległości HC-SR04 jest podłączony

do portów cyfrowych o numerach 4 (Trig) oraz 5 (Echo). Drugi czujnik jest podłączony pod portów o numerach 6 (Echo) oraz 7 (Trig). Oba są zasilane takim samym napięciem 5V. Moduł bluetooth jest podłączony portem TX do portu cyfrowego o numerze 0 (RX), natomiast portem RX do portu o numerze 1 (TX), a zasilany jest napięciem 5V. Akcelerometr GY-291 jest podłączony do zasilania 5V. Pin SCL jest podłączony do pinu analogowego o numerze 5, natomiast pin SDA jest podłączony do pinu analogowego o numerze 4.

### 4. Moduł programistyczny

Poniżej przedstawiono diagram przypadków użycia (DPU), który prezentuje główne funkcjonalności opracowanego rozwiązania.

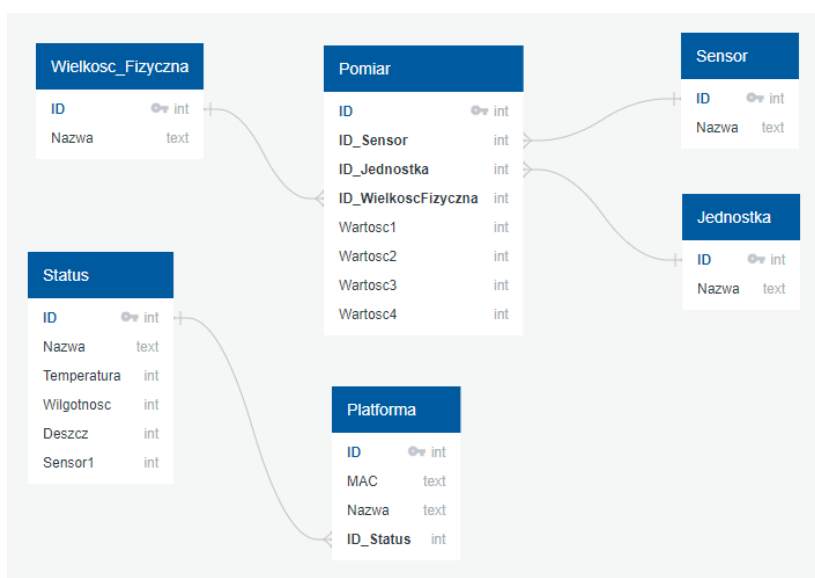


Rysunek 2. Diagram przypadków użycia

Powyższy diagram przedstawia przypadki użycia aplikacji. Składa się ona z trzech modułów: Aplikacja Android, Arduino oraz Baza danych SQLite. Każdy z tych systemów posiada funkcjonalności przedstawione w zielonych elipsach. Strzałki wskazują powiązania jakie istnieją między tymi funkcjami. Dla przykładu po uruchomieniu programu przez użytkownika oraz

do dodaniu nowej kolekcji czujników, funkcja ta odwołuje się do funkcji Dodawanie nowego wpisu z kolekcją czujników w systemie Baza danych SQLite.

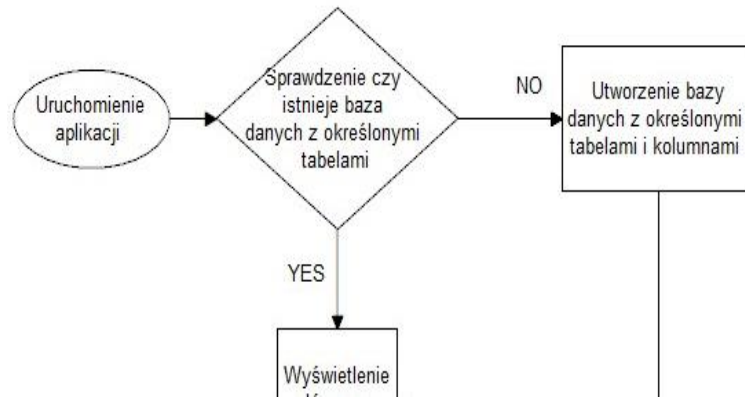
Na potrzeby praktycznej realizacji projektu oraz przeprowadzenia testów rozwiązania, zaprojektowano i implementowano bazę danych, której model przedstawiony jest poniżej.



Rysunek 3. Model relacyjny bazy danych.

Baza danych programu składa się z sześciu tabel. Tabele Wielkosc\_Fizyczna, Sensor oraz Jednostka składają się z dwóch kolumn ID oraz Nazwa, gdzie pierwsza z nich jest kluczem głównym i są w relacjach jeden do wielu z tabelą Pomiar. Tabela Pomiar zawiera klucz główny ID, klucze obce ID\_Sensor, ID\_Jednostka, ID\_WielkoscFizyczna oraz kolumny Wartosc1, Wartosc2, Wartosc3 i Wartosc4 przechowujące wartości odczytów

z czujnika. Dwie ostatnie tabele to Status, która zawiera informacje o kolekcjach czujników oraz Platforma, która zawiera informacje o urządzeniach. Kolumna ID w tabeli Status jest kluczem głównym połączonym z tabelą Platforma w relacji jeden do wielu z kluczem obcym ID\_Status. Na poniższych schematach przedstawiono opracowane algorytmy, które służą do obsługi bazy danych oraz związanych z komunikacją z bazą funkcjonalności.



Rysunek 4. Schemat blokowy tworzenia baz danych



```

blSocket.getOutputStream().write("a".getBytes());
try {
    Thread.sleep( millis: 1000);
}
catch (InterruptedException e) { }
InputStream=blSocket.getInputStream();
int byteCount = inputStream.available();
if(byteCount > 0) {
    byte[] rawBytes = new byte[byteCount];
    inputStream.read(rawBytes);
    final String string = new String(rawBytes, charsetName: "UTF-8");
    temperature = string;
    tbTemperature.setText(temperature);
}

```



Rysunek 7. Schemat blokowy odczytu pomiarów

Kod ten przedstawia jakie działania są podejmowane przez aplikację po wciśnięciu przycisku “Read”, który ma za zadanie pobranie odczytów z urządzenia Arduino. Wysyłane są poprzez Bluetooth wartości “a”, “b”, “c” oraz “d”, które odpowiadają za wysłanie przez Arduino do aplikacji odpowiedniego odczytu, to znaczy dla wartości “a” jest to odczyt z czujnika temperatury, dla “b” jest to odczyt z czujnika wilgotności, dla “c” jest to czujnik deszczu, natomiast dla “d” jest to sensor1, który jest miejscem na dodanie przez użytkownika dodatkowego wybranego czujnika. Dane są wysyłane w formacie “byte” do Arduino, a urządzenie przesyła dane do aplikacji także w formacie “byte”, co jest następnie konwertowane do formatu string z kodowaniem “UTF-8”. Na samym końcu otrzymana wartość jest przypisywana do textbxa odpowiadającego danemu czujnikowi.w ramach zestawu.

## Literatura

1. Jaskot A., Posiadała B. Analysis of motion of the three wheeled mobile platform. MATEC Web of Conferences 157, 01008, 2018.
2. Tao Song, Bang-Guo Wei A calibration method of dual two-dimensional laser range finders for mobile manipulator., Intelligent Manufacturing and Robotics – Research Article 2019.
3. Rodriguez K., Crespo J, Barber R. An Android Interface for an Arduino Based Robot for Teaching in Robotics, The 6th International Conference of Education, Research and Innovation, Sevilla 2013
4. Simon Monk, Arduino : 36 projektów dla pasjonatów elektroniki, Wydawnictwo Helion, 2015.
5. Android Studio: tworzenie aplikacji mobilnych, Marcin Płonkowski, Helion, 2018.



# PROJEKT SYSTEMU MONITOROWANIA OBTURACYJNEGO BEZDECHU SENNEGO

Przemysław Szyszkowski, Grzegorz Śmigieński

Uniwersytet Kazimierza Wielkiego  
Instytut Informatyki  
ul. Kopernika 1 p.204, 85-074 Bydgoszcz  
e-mail: p\_szyszkowski@wp.pl, gsmigielski@ukw.edu.pl

*W artykule przedstawiono prototyp urządzenia, opartego o mikrokontroler AVR, służącego do monitorowania obturacyjnego bezdechu sennego.*

**Słowa kluczowe:** *Obturacyjny bezdech senny, mikrokontroler, baza danych, aplikacja webowa, aplikacja mobilna*

## Design of an obstructive sleep apnea monitoring system

**Abstrakt:** *Article presents prototype of device based on microcontroller AVR, which serves as an obstructive sleep apnea monitor.*

**Keywords:** *Obstructive sleep apnea, microcontroller, database, Web application, mobile application*

### 1. Wprowadzenie

Obturacyjny bezdech senny jest chorobą, której przyczyna leży w epizodach zatrzymania oddechu (bezdechach) powtarzających się wielokrotnie w czasie snu. Za bezdech uważa się sytuację, w której wentylacja płuc zostaje wstrzymana na okres dłuższy niż 10 sekund lub zachodzi spływanie oddechu poniżej 50%. Fakt ten prowadzi do obniżenia wysycenia krwi tlenem. U osób cierpiących na zespół bezdechu sennego zwykle rejestruje się dziesiątki, a nawet setki takich epizodów w czasie jednej nocy. Najczęściej występującym rodzajem bezdechu sennego, obejmującym ponad 80% przypadków, jest obturacyjny bezdech senny. Jego przyczyną są ponawiające się zatrzymania przepływu powietrza w obszarze górnych dróg oddechowych (gardło, nos), skutecznie tamujących oddech, mimo wzmożonej pracy płuc. Zazwyczaj bezpośrednim powodem niemożności płynnego oddechu jest zwiotczenie mięśni gardła i języka. W wyniku rozluźnienia i opadnięcia mięśni podniebienia miękkiego oraz języczka, następuje blokowanie przepływu powietrza. Stan ten prowadzi do wysiłonego i głośnego oddechu (chrapania), a następnie do zupełnego zatrzymania oddechu. Wśród objawów, poza chrapaniem i obserwowanymi przez partnerkę/partnera chorego przerwami w oddychaniu w czasie snu, wyróżnia się inne, charakterystyczne dolegliwości: poranne bóle głowy, problemy z

koncentracją, zaburzenia pamięci, permanentne zmęczenie, zasypianie w ciągu dnia. Obturacyjnemu bezdechowi sennemu często towarzyszą inne schorzenia, takie jak otyłość, zaburzenia rytmu serca czy nadciśnienie tętnicze. Informacje te pokazują jak ważne jest wczesne zdiagnozowanie problemu i podjęcie skutecznego leczenia. [1, 2, 3] W pracy opisano prototyp urządzenia, które umożliwia samodzielne (w warunkach domowych) zdiagnozowanie bezdechu sennego.

### 2. Istniejące rozwiązania rynkowe

Na rynku urządzeń medycznych, można wyróżnić kilka rozwiązań dedykowanych diagnostyce obturacyjnego bezdechu sennego. W zależności od ceny, która waha się od kilku do kilkudziesięciu tysięcy złotych, dostępne urządzenia, umożliwiają monitorowanie takich parametrów jak wysiłek oddechowy, tętno, nasycenie tlenem, jak również przepływ powietrza przez nos i chrapanie. Droższe pozwalają na zapis sygnałów EEG, EKG, GSR, EMG. Posiadają też możliwość bezprzewodowej komunikacji z sensorami oraz dodatkowym urządzeniem (smartfon, tablet).

### 3. Elementy składowe systemu

Zaprojektowany system składa się z czterech współpracujących ze sobą elementów. Pierwszym z nich jest urządzenie (TeleOBS)

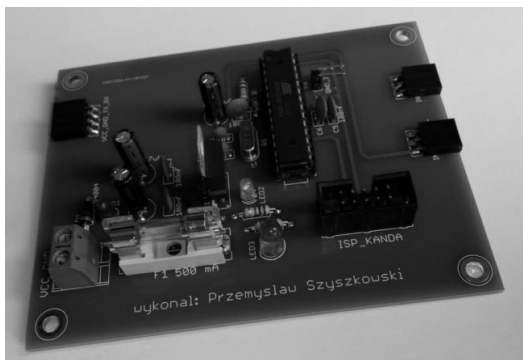
w postaci sterownika mikroprocesorowego. Drugi element systemu stanowi aplikacja mobilna na system Android, która poprzez transmisję Bluetooth ze sterownikiem TeleOBS odpowiada za graficzną reprezentację badanych parametrów oraz za ich eksport do bazy danych. Baza danych, będąca trzecim elementem systemu, pozwala na gromadzenie wyników przeprowadzanego badania na zdalnym serwerze. Ostatni element systemu - serwis webowy, jest narzędziem przeznaczonym dla lekarzy kierujących pacjentów na badanie diagnostyczne.

#### 4. Konstrukcja urządzenia

Sterownik został oparty o popularny mikrokontroler 8-bitowy z rodziny AVR - ATmega8. W skład urządzenia wchodzi płytkę z mikrokontrolerem (rys. 1) oraz zestaw modułów zewnętrznych, podłączanych do sterownika.

W projekcie został wykorzystany moduł Bluetooth XM-15B używający do komunikacji interfejsu UART (RX, TX). Maksymalny zasięg działania modułu wynosi 30 m, co stanowi spełnienie oczekiwań wobec systemu funkcjonującego w mieszkaniu czy domu jednorodzinnym (funkcja wizualnego monitoringu czynności oddechu i pulsu na ekranie smartfonu/tabletu).

Jako czujnik tętna ludzkiego serca zastosowany został optoelektroniczny sensor Grove. Pozwala on na wykrycie uderzeń serca wskutek zmiany parametrów oświetlonych tkanek. W skład zestawu wchodzi przewodowy sensor z klipsem oraz moduł odbiornika. Podczas pomiaru klips może się znajdować zarówno na uchu jak i na palcu, a jego przewód o długości 120 cm pozwala na wygodę użytkownika bez krępowania ruchów pacjenta.



Rysunek 1. Płytkę PCB sterownika

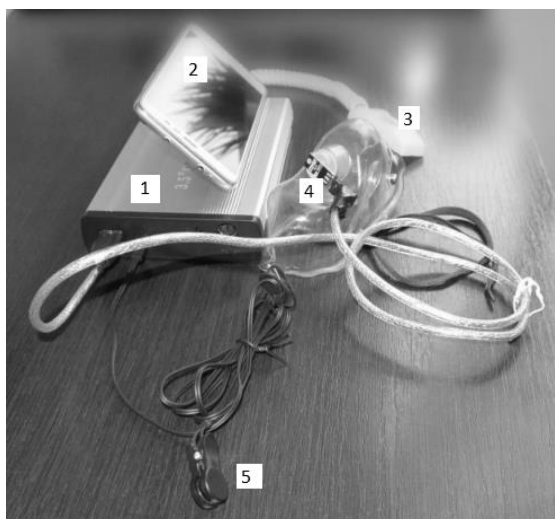
Jako czujnik oddechu, wykorzystano moduł mikrofonu z wbudowanym wzmacniaczem LMV321L o nazwie handlowej KAmoMIC\_ELECTRET. Moduł ten

wykorzystuje mikrofon elektretowy typu KPCM6B. Wbudowany wzmacniacz zapewnia 100-krotne wzmocnienie sygnału, dzięki czemu możliwe jest mierzenie napięcia na wyjściu, np. poprzez wykorzystanie przetwornika ADC w mikrokontrolerze. Wybrany moduł pozwala na zaimplementowanie w projekcie funkcji reagowania na dźwięk. Fakt ten został wykorzystany w urządzeniu TeleOBS do identyfikacji oddechu pacjenta. Zamontowanie czujnika w masce tlenowej potęguje natężenie szmeru wynikającego z oddychania, dzięki czemu czujnik wykrywa oddech, sygnalizując ten fakt zmianą napięcia na wyjściu.

Wszystkie kody źródłowe mikrokontrolera zostały napisane w języku C, z wykorzystaniem kompilatora AVR-GCC, w środowisku Eclipse [4]. Funkcje programu:

- kontrola i przesyłanie informacji o stanie pulsu pacjenta na podstawie czujnika podłączonego do jednego z wejść (obsługa przerw zewnętrznych),
- kontrola i przesyłanie informacji o stanie oddechu pacjenta na podstawie czujnika podłączonego do wejścia mikrokontrolera (konwersja analogowo-cyfrowa),
- przesyłanie informacji o wciśnięciu przycisku alarmowego ("emergency"),
- obsługa bezprzewodowej transmisji Bluetooth poprzez interfejs USART,
- kontrola poprawności wykonywanego programu i zabezpieczenie przed zawieszeniem urządzenia (Watchdog).

Wybór obudowy był podyktowany względami estetycznymi jak i ekonomicznymi. Z uwagi na wysokie ceny obudów wykonanych z profili aluminiowych, zdecydowano się na wykorzystanie zewnętrznej obudowy dysku twardego 3,5", po jej wcześniejszej adaptacji. Obudowa tego typu nie odbiega niczym od dedykowanych rozwiązań wspomnianych wyżej, a jej niska cena stanowi argument potwierdzający zasadność wyboru. Ponadto, zdecydowano się na użycie uchwyty magnetycznego do telefonu/tabletu, który zamocowano na wierzchniej stronie obudowy. Dzięki temu ekran monitorujący parametry pacjenta można bez problemu zamontować na obudowie, jak i go zdemontować celem zdalnej obserwacji.



**Rysunek 2** Widok zmontowanego urządzenia: 1 - obudowa, 2 - smartfon, 3 - przycisk alarmowy "emergency", 4 - maska z czujnikiem oddechu, 5 - czujnik pulsu.

## 5. Oprogramowanie do monitorowania bezdechu sennego

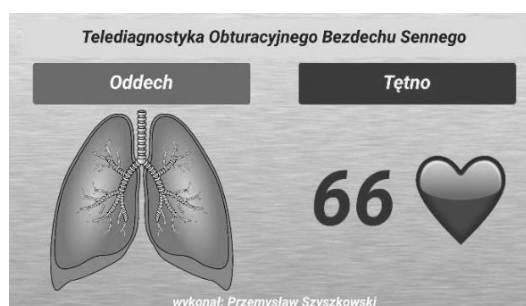
Istotnym elementem systemu monitorowania bezdechu sennego jest aplikacja mobilna przeznaczona dla pacjenta lub osób monitorujących czynności życiowe pacjenta w warunkach domowych. Aplikacja powstała z wykorzystaniem narzędzia AppInventor, dedykowanego do tworzenia oprogramowania na system Android [5].

Cechy aplikacji:

- interfejs przyjazny użytkownikowi,
- łączność ze sterownikiem TeleOBS dzięki bezprzewodowej transmisji Bluetooth,
- graficzna i cyfrowa prezentacja parametrów związanych z oddechem i pulsem pacjenta,
- ciągle monitorowanie obustronnej transmisji Bluetooth i sygnalizowanie stanu zerwanego połączenia ze sterownikiem (autodiagnostyka),
- automatyczne testowanie łączności z siecią Internet i bazą danych przed umożliwieniem rozpoczęcia badania (autodiagnostyka),
- odbiór, interpretacja i przetwarzanie sygnałów pochodzących ze sterownika OBS (oddech, puls, wezwanie pomocy),
- realizacja wpisów do bazy danych z informacją o dacie i godzinie rozpoczęcia oraz zakończenia badania,
- cykliczne uruchamianie skryptu PHP, realizującego wstawienie nowego rekordu do bazy danych z informacją o aktualnym pulsie pacjenta,
- uruchamianie skryptu PHP odpowiedzialnego za wpis do bazy danych, w przypadku wykrycia epizodu bezdechu sennego,

- funkcja wstrzymania trwającego badania na żądanie pacjenta (konieczność zdjęcia maski tlenowej), pozwalająca na czasowe ignorowanie sygnałów ze sterownika TeleOBS, które fałszowałyby wynik w związku z nieobecnością pacjenta na stanowisku diagnostycznym.

Na rys. 3 przedstawiono ekran aplikacji podczas normalnej pracy urządzenia, w przypadku wykrytego oddechu i pulsu - czerwony kolor grafiki płuc oraz obecność ikony serca. Wykrycie bezdechu jest sygnalizowane zmianą koloru grafiki płuc, a brak pulsu - zniknięciem ikony serca.



**Rysunek 3.** Ekran aplikacji podczas normalnej pracy.

Aplikacja mobilna, z którą styczność będzie miał pacjent, musi być intuicyjna i nieprzysparzająca problemów użytkownikowi, niezależnie od jego wieku. Jednocześnie pełni ona kluczową funkcję monitoringu diagnostycznego, z jednej strony nawiązując łączność ze sterownikiem i interpretując wysyłane przez niego dane, a z drugiej odpowiadając za ich prezentację oraz eksport do bazy danych. Utworzona baza danych z kolei, pozwala na nieulotne przechowywanie parametrów przeprowadzanego badania, takich jak bieżący puls pacjenta, zarejestrowane epizody bezdechu sennego, a także czas trwania diagnostyki. Serwis webowy przeznaczony dla lekarzy, umożliwi rejestrację pacjenta, podgląd aktualnego stanu trwającego badania jak i wgląd do kompletnych zestawień reprezentujących wyniki przeprowadzonych czynności diagnostycznych.

Z uwagi na wysoką dostępność i brak dodatkowych opłat, zdecydowano się na wykorzystanie systemu bazodanowego jakim jest MySQL [6]. System ten dostępny jest u wielu przedstawicieli usług hostingowych i jest bezpłatny, co czyni go szczególnie atrakcyjnym. Baza danych w wersji testowej powstała z wykorzystaniem oprogramowania XAMPP, pozwalającego na lokalną instalację serwera bazy danych. Ostateczna wersja utworzonej bazy danych, została umieszczona na zewnętrznym serwerze (cba.pl).

Wizualna strona serwisu została opracowana z wykorzystaniem hipertekstowego języka znaczników (HTML) oraz kaskadowych arkuszy stylów (CSS). W celu wygodnego zarządzania wyglądem wszystkich plików składających się na opracowany serwis, stworzono plik „style.css”, w którym ustawione zostały globalne zasady formatowania całej witryny. Strona internetowa została opublikowana na zewnętrznym serwerze. [6, 7, 8] Na rys. 4. przedstawiono panel główny serwisu webowego dostępnego po zalogowaniu.



**Rysunek 4.** Panel główny serwisu webowego.

W wyniku połączenia języka PHP i systemu bazodanowego MySQL, utworzono oprogramowanie, w którym można wyróżnić następujące funkcje:

- system logowania,
- brak możliwości przejścia do innych podstron serwisu (np. poprzez wpisanie nazwy pliku podstrony w pasku adresu przeglądarki) w przypadku niezalogowanego użytkownika,
- identyfikacja lekarza (imię i nazwisko) w systemie na podstawie numeru identyfikacyjnego, podawanego podczas logowania,
- wykaz wszystkich zarejestrowanych pacjentów (pozyskiwanie danych z bazy),
- formularz rejestracji pacjenta (wprowadzanie nowych rekordów do bazy),
- wyszukiwarka wyników badań (wyszukiwanie danych w bazie),
- usuwanie pacjenta (usuwanie rekordów z bazy danych),
- wylogowanie z systemu (zakończenie sesji).

## 6. Wnioski

System testowany był na dziesięciu osobach w wieku od 10 do 70 lat i w każdym przypadku radził sobie bezbłędnie z identyfikacją oddechu oraz pomiarem pulsu. Ponadto, wszyscy użytkownicy aplikacji uznali ją za łatwą w obsłudze i przyjazną. Serwis webowy służący do obsługi pacjenta, wśród osób testujących zyskał miano intuicyjnego.

## Literatura

1. Zieliński J., Pływaczewski R., Bednarek M. Zaburzenia oddychania w czasie snu, PZWL, Warszawa, 2006.
2. Epstein L. J., Kristo D., Strollo P. J. i wsp. Clinical guideline for the evaluation, management and long-term care of obstructive sleep apnea in adults. *J. Clin. Sleep Med.* 2009; 5:263–276.
3. Garvey J. F., Pengo M. F., Drakatos P., Kent B. D. Epidemiological aspects of obstructive sleep apnea. *J. Thorac. Dis.* 2015; 7:920–929.
4. Williams E. Programowanie układów AVR dla praktyków, Wydawnictwo Helion, Gliwice, 2014.
5. Wolber D., Abelson H., Spertus E. Looney L. App Inventor 2. Create Your Own Android Apps. 2nd Edition, Wydawnictwo O’Reilly Media, 2014.
6. Rockoff L. Język SQL. Przyjazny podręcznik. Wydanie II (ebook), Wydawnictwo Helion, 2017.
7. Duckett J. HTML i CSS. Zaprojektuj i zbuduj witrynę WWW., Wydawnictwo Helion, Gliwice, 2014.
8. Lis M. PHP i MySQL. Dla każdego. Wydanie III (ebook), Wydawnictwo Helion, 2017.

# ROZPROSZONE SYSTEMY BAZ DANYCH W PLANOWANIU PRODUKCJI

Izabela Rojek<sup>\*1</sup>, Zbyszko Królikowski<sup>2</sup>, Mariola Marciniak<sup>3</sup>

<sup>1</sup> Uniwersytet Kazimierza Wielkiego, Instytut Informatyki, Chodkiewicza 30, 85-064 Bydgoszcz

<sup>2</sup> Politechnika Poznańska, Wydział Informatyki, Pl. Marii Skłodowskiej-Curie 5, 60-965 Poznań

<sup>3</sup> Uniwersytet Kazimierza Wielkiego, Instytut Matematyki, Chodkiewicza 30, 85-064 Bydgoszcz  
e-mail: izarojek@ukw.edu.pl

**Streszczenie:** We współczesnym globalnym świecie wiele przedsiębiorstw zmienia swoją strategię zarządzania przedsiębiorstwem opartą na scentralizowanej bazie danych na strategię opartą na rozproszonych systemach baz danych. W artykule scharakteryzowano rozproszone systemy baz danych w planowaniu produkcji. W szczególności omówiono cechy planowania produkcji oraz zdefiniowano rodzaje rozproszonych systemów baz danych. Jako przypadek szczególny omówiono możliwość zastosowania rozproszonych systemów baz danych w planowaniu produkcji w wybranym przedsiębiorstwie produkcyjnym.

**Słowa kluczowe:** Rozproszone systemy baz danych, przedsiębiorstwo, planowanie produkcji

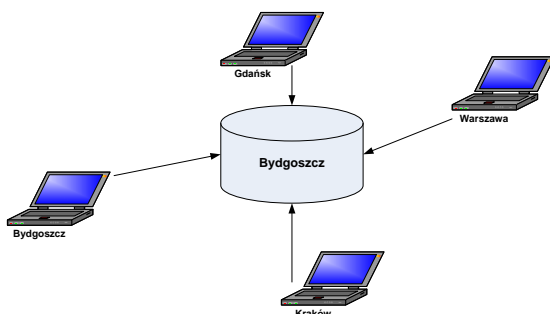
## *Distributed databases systems in production planning*

**Abstract:** In the modern global world, many enterprises are changing their enterprise management strategy based on a centralized database to a strategy based on distributed database systems. The article characterizes distributed database systems in production planning. In particular, the features of production planning were discussed and the types of distributed database systems were defined. As a special case, the possibility of using distributed database systems in production planning in a selected enterprise was discussed.

**Key words:** Distributed databases systems, enterprise, production planning

## 1. Wprowadzenie

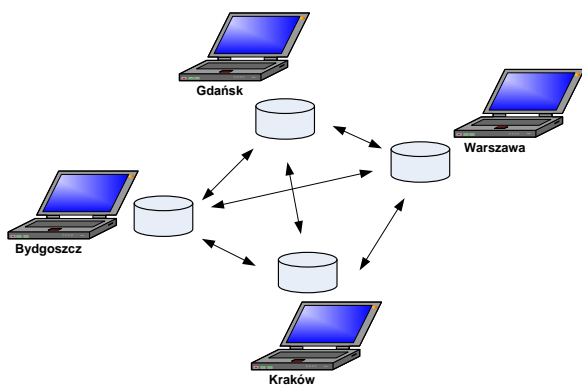
Dane gromadzone i przetwarzane przez przedsiębiorstwa są najczęściej przechowywane w systemach informatycznych opartych na bazach danych. W typowych zastosowaniach systemów baz danych wykorzystuje się architekturę scentralizowaną, w której system zarządzania bazą danych oraz wszystkie dane znajdują się w jednym węźle sieci informatycznej (rys. 1) [1,2,3].



Rysunek. 1 Scentralizowana baza danych

Istnieje jednak wiele rozwiązań, w których scentralizowane bazy danych nie zapewniają wymaganej funkcjonalności i efektywności pracy [4].

Wielkie, korporacje, międzynarodowe firmy, często nawet o zasięgu globalnym, w poszukiwaniu zysku dążą do jak najlepszego zaspokajania potrzeb wszystkich swoich klientów. Potrzeby te mogą być diametralnie różne nie tylko na różnych kontynentach, ale nawet na obszarze jednego państwa. W związku z tym wybierane są takie systemy, które gwarantują podejmowanie decyzji w lokalnych oddziałach firm, ale jednocześnie umożliwiają dostarczanie jak najpełniejszej informacji zarządzanej centralnie po to, aby ich działania były jak najbardziej efektywne. Naturalnym wyborem w takich sytuacjach są rozproszone systemy baz danych (rys. 2).



**Rysunek. 2** Rozproszone systemy baz danych

Gdyby zbudować system informatyczny dla takiego przedsiębiorstwa, opartego na scentralizowanej bazie danych umieszczonej np. w Bydgoszczy, wówczas każde odwołanie do tej bazy z innego miasta wymagałoby transmisji sieciowej. Przy sieci o niskiej przepustowości i dużej częstotliwości odwołań poprawne wykorzystywanie systemu stałoby się niemożliwe. Dodatkowo, taki system byłby znacznie bardziej podatny na awarie niż system rozproszony. Rozwiązaniem alternatywnym do rozwiązania scentralizowanego jest zastosowanie wielu lokalnych baz danych, rozmieszczonych w każdym oddziale przedsiębiorstwa.

## 2. Planowanie produkcji

Centralnym procesem każdego systemu zarządzania produkcją jest planowanie, którego zadaniem staje się wygenerowanie właściwego planu oraz rozłożenie go w czasie [5]. Planowanie produkcji to złożony proces. Obejmuje zarówno tworzenie planów strategicznych, taktycznych jak i operacyjnych. Jego nieodłączną częścią jest podejmowanie decyzji w oparciu o posiadane informacje. Plan strategiczny obejmuje misję oraz cele strategiczne przedsiębiorstwa, analizę otoczenia, strategię ogólną i strategię funkcjonalne. W przedsiębiorstwie produkcyjnym planowanie strategiczne obejmuje następujące obszary: strategię produktu, zagregowany plan produkcji, strategię rozwoju zdolności produkcyjnych i inwestycji, strategię finansową oraz strategię pracy i zatrudnienia. Podstawą do tworzenia planów strategicznych są badania marketingowe oraz prognozy (ekonomiczne, techniczne, zapotrzebowania i sprzedaży). Podstawowym zadaniem planowania taktycznego jest obliczanie planów produkcji poszczególnych wyrobów na podstawie planów produkcji łącznej oraz ich

podziału w czasie (np. rozbitcie planów miesięcznych na tygodniowe). Planowanie taktyczne obejmuje w przedsiębiorstwie następujące działy: badanie, rozwój i przygotowanie produkcji, obsługę serwisową, finanse oraz dystrybucję. Na poziomie planowania taktycznego tworzone są plany: taktycznego zapotrzebowania materiałowego (MRP I), taktycznego zapotrzebowania na zasoby produkcyjne (MRP II), sterowania jakością produktów, utrzymania ruchu i niezawodności urządzeń oraz sieci i instalacji produkcyjnych. Planowanie taktyczne stanowi wytyczne dla planów operacyjnych. Planowanie operacyjne jest to krótkookresowy plan produkcji, którego podstawą jest rozdział zadań wytwórczych oraz gospodarka zapasami. Rozdział zadań produkcyjnych sprowadza się do wyznaczenia konkretnych operacji, terminów ich realizacji oraz wykonawców. Planowanie operacyjne polega na: przydziale zleceń produkcyjnych, dostarczaniu wyposażenia narzędziowego, materiałów i instrukcji wykonywanych, zapewnieniu obsługi poszczególnych stanowisk pracy, ustaleniu kolejności wykonywanych zadań, określeniu terminu rozpoczęcia i zakończenia zadań, badaniu jakości wykonania oraz aktualizacji stanu zaawansowania prac.

## 3. Rozproszone systemy baz danych

Szybki rozwój technologii i spadek cen nośników danych spowodował gromadzenie dużych ilości danych niemal w każdej dziedzinie życia, w szczególności w sferze gospodarki. Pojawia się naturalne pytanie o możliwość wykorzystania danych zgromadzonych przez przedsiębiorstwo w celu poprawienia jego rentowności. Duże firmy posiadające swoje oddziały w wielu lokalizacjach geograficznych używają w swojej działalności rozproszonego systemu baz danych czyli *zbioru lokalnych baz danych stanowiących całość w sensie jednego modelu danych i koordynacji wykonywanych transakcji* [4]. Rozwiązanie takie pozwala na budowę sprawnego systemu wspomagania decyzji zarówno w lokalnych oddziałach firmy jak i dostęp do pełnej informacji zarządzanej centralnie.

Operacje wykonywane przez lokalną bazę danych są kontrolowane autonomicznie przez każdy z węzłów. Integrację oraz przezroczysty dostęp, czyli taką organizację lokalnych baz danych aby z punktu widzenia użytkownika końcowego stanowiły one logicznie jedną zintegrowaną bazę danych realizuje oprogramowanie sieciowe oraz mechanizmy dostępu do baz zdalnych. Stosuje się obiekty zwane łącznikami pozwalające wywoływać

procedury bazy zdalnej. Używane są zarówno łączniki publiczne, dostępne dla wszystkich użytkowników jak i prywatne będące własnością użytkownika, który go utworzył. Można również definiować łączniki bazodanowe, które podczas podłączania do zdalnej bazy danych uwierzytniają bieżącego użytkownika, który jest aktualnie podłączony do lokalnej bazy danych. Używa się zaawansowanych metod replikacji danych, tak aby użytkownik operował w ten sam sposób na danych oryginalnych jak i na ich kopii. Powstała w wyniku replikacji redundancja z jednej strony poprawia bezpieczeństwo danych, które są dostępne nawet w razie awarii jednego z węzłów, z drugiej strony powoduje konieczność wprowadzenia mechanizmów zapewniających spójność danych. System powinien zapewniać zarządzanie transakcjami odwołującymi się do wielu węzłów systemu. Transakcje takie muszą posiadać cechy trwałości, spójności, atomowości i izolacji tak, jak to ma miejsce w przypadku transakcji w scentralizowanych bazach danych. Zatwierdzanie transakcji ma charakter dwuetapowy – wyróżniony węzeł zwany koordynatorem transakcji jest informowany przez bazy lokalne czy lokalna część transakcji ma być zatwierdzona czy wycofana, na ostateczne zatwierdzenie zmian muszą się zgodzić wszystkie bazy lokalne [2,4].

Systemy rozproszone charakteryzują się pewnymi, sobie tylko właściwymi cechami. Są to:

- konieczność jednoznacznej identyfikacji każdego spośród jego zasobów,
- rozproszony charakter sterowania – każdy węzeł powinien dysponować programowo-sprzętowymi możliwościami realizacji funkcji sterowania siecią, przy czym zwiększone wymagania w stosunku do niezawodności systemu rozproszonego prowadzą do konieczności unikania takich algorytmów sterowania, które bazowałyby na wydzieleniu węzłów uprzywilejowanych w jakiś sposób w porównaniu z innymi węzłami,
- niejednorodność (ang. *heterogeneity*) węzłów sieci – mogą się one różnić np. formatem przedstawienia informacji, oprogramowaniem systemowym i użytkowym.

Próby efektywnego zaspokojenia potrzeby dostarczania konkretnych danych we właściwe miejsce powodują wybranie jednej z kilku dostępnych technik:

1. Pierwsza z nich polega na tym, że poszczególne fragmenty bazy znajdują się tylko w konkretnych miejscach, natomiast replikowane są jedynie katalogi zawierające informację o położeniu poszczególnych fragmentów. Zalety: brak konieczności synchronizacji wielu kopii oraz zmniejszenie ilości przesyłanych informacji. Wada: trudności w przypadku pracy z dużą ilością danych jednocześnie.

2. Kolejna technika – replikowana baza danych – polega na kopiowaniu całej bazy w jedno lub kilka wyznaczonych miejsc. Dzięki takiemu rozwiązaniu dostęp do danych jest lokalny i unika się konieczności decydowania, jakie fragmenty bazy i gdzie mają zostać skopiowane. Wada: niska efektywność z powodu konieczności składowania ogromnych baz w kilku miejscach oraz dłuższe wyszukiwanie odpowiedzi.
3. Replikowanie na poziomie tabel to następna technika rozpraszania danych, polegająca na kopiowaniu tylko niezbędnych tabel, bez modyfikacji zawartych w nich danych. Stosowana jest tam, gdzie ważne jest oddzielenie danych operacyjnych od aplikacji wspomagających podejmowanie decyzji. Zabronienie użytkownikom uruchamiania zapytań do aktywnych tabel bazy zapobiega konfliktom jednoczesnego dostępu. Trudności mogą pojawić się przy rekonstrukcji logicznej całości bazy danych po ewentualnej awarii systemu.
4. Innym sposobem dystrybucji danych jest tzw. podział danych. Polega on na tym, że dokonuje się podziału i replikacji logicznej tabeli do wielu punktów.
5. Ostatnią techniką jest dystrybucja fan-out, która jest próbą rozwiązania problemu wzrastającej – wraz ze zwiększeniem liczby punktów dystrybucji – trudności zarządzania bazą, co z kolei zwiększa podatność systemu na błędy. Sposób ten polega na wprowadzeniu kilku pośrednich punktów dystrybucji, dzięki czemu nie trzeba utrzymywać połączenia pomiędzy każdym magazynem danych i każdą zreplikowaną stroną, a wystarczy jedynie utrzymywać połączenia z głównego systemu danych do regionalnych centów przetwarzania, skąd łatwiej jest zarządzać połączeniami do oddziałów terenowych. Dzięki temu ogranicza się liczbę połączeń o dużej szybkości i wysokim koszcie przesłania informacji do punktów dystrybucji [6].

Podstawowymi typami rozproszonych baz danych są: jednorodne rozproszone bazy danych, niejednorodne rozproszone bazy danych oraz federacyjny system baz danych. Często rozproszone systemy baz danych zawierają wiele serwerów. Jeśli wszystkie węzły posiadają systemy zarządzania bazą danych pochodzące od jednego producenta mówić możemy o jednorodnej rozproszonej bazie danych. W niejednorodnej bazie danych poszczególne węzły posiadają różne konfiguracje sprzętowe i oprogramowania. Powoduje to konieczność stosowania bramek (gateways). Federacyjne systemy baz danych składają się z pewnej ilości względnie niezależnych systemów baz

danych. Istnieje przy tym możliwość, aby niekiedy część z nich wykonywała wspólnie pewne ustalone funkcje [1,2,4].

Rozróżnia się:

- jednorodne rozproszone systemy baz danych,
- niejednorodne rozproszone systemy baz danych,
- federacyjne systemy baz danych.

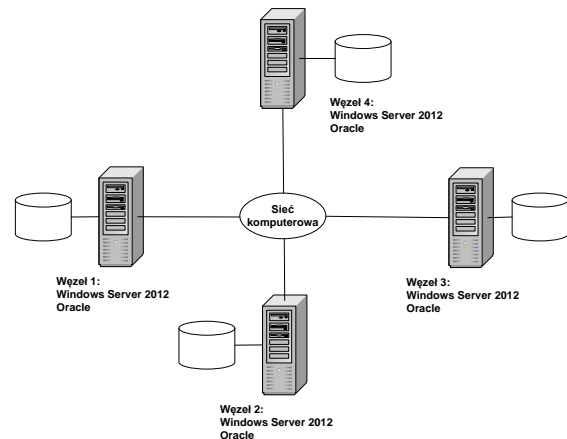
#### Jednorodne rozproszone systemy baz danych

Systemy jednorodnych baz danych mają następujące cechy:

- system Zarządzania Bazami Danych (SZBD) tego samego producenta we wszystkich węzłach,
- możliwość użycia wbudowanych rozwiązań dla rozproszonych baz danych.

W jednorodnej rozproszonej bazie danych dane są rozłożone między dwa lub więcej systemów, każdy oparty na tym samym rodzaju systemu zarządzania bazą danych (np. ORACLE). Na ogół taki rozproszony system działa na tego samego rodzaju sprzęcie pod tym samym systemem operacyjnym (rys. 3).

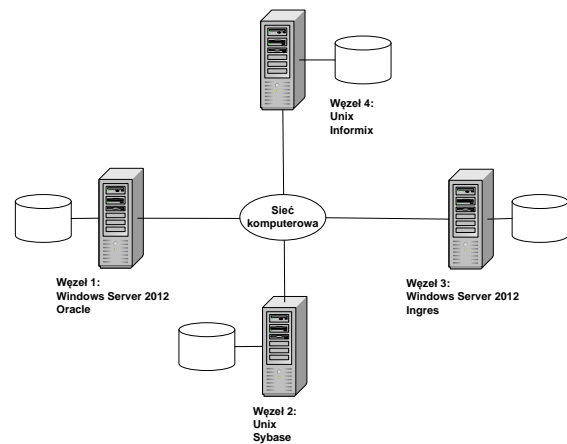
cel do osiągnięcia, nad którym pracuje wielu producentów systemów baz danych.



**Rysunek. 3** Przykład jednorodnego systemu baz danych

#### Niejednorodne rozproszone systemy baz danych

W niejednorodnej rozproszonej bazie danych konfiguracje sprzętowe i oprogramowania są różne. W jednym miejscu może być ORACLE działający pod Windows Server 2012, w drugim Sybase pod UNIX, a w jeszcze innym Ingres pod Windows Server 2012. Pokazano to na rysunku 4. Obecnie podstawowym sposobem uzyskiwania niejednorodnego systemu jest łącze (gateway). Łącze jest interfejsem z jednego SZBD do drugiego, zwykle dostarczany przez konkretnego producenta SZBD.



**Rysunek. 4** Przykład niejednorodnego systemu baz danych

#### Federacyjne systemy baz danych

Federacyjne bazy danych mają następujące cechy:

- względnie niezależne bazy danych,
- niekiedy używane łącznie do wykonania pewnych funkcji,
- z ograniczonym dostępem z zewnątrz poprzez pewien podschemat,
- połączenia między bazami danych typu 1-1.

Federacyjne, czasem nazywane wielobazowymi, systemy rozproszonych baz danych przypominają polityczny model federacji. Federacyjne systemy baz danych składają się z pewnej liczby względnie niezależnych autonomicznych baz danych. Niekiedy zachodzi potrzeba zebrania części lub wszystkich z tych oddzielnych baz danych, aby wykonać wspólną funkcję. Niektóre aspekty federacji zaczynają się krystalizować z uwzględnieniem połączenia otwartych systemów i standardu dostępu do odległych baz danych. Stanowią one jednak ciągle

Od rozproszonych systemów baz danych oczekuje się, że reprezentują one jeden model danych w przedsiębiorstwie. Zasadniczym celem systemów bazy danych jest to, aby dla użytkownika wyglądała ona jak jedna, scentralizowana baza danych. Innymi słowy system rozproszonej bazy danych powinien mieć trzy rodzaje przezroczystości:

##### 1. Przezroczystość lokalizacji

Użytkownicy muszą wiedzieć, w którym dokładnie miejscu są przechowywane dane. Zatem kierownik, który pragnie się dowiedzieć ilu pracowników zatrudnia biuro w np. Krakowie, nie musi być świadomy tego, że ma do czynienia z rozproszoną bazą danych. Zaletą przezroczystości lokalizacji jest to, że upraszcza programy użytkownika i interfejsu. Dane mogą migrować między lokalizacjami, nie powodując błędów w żadnym z tych programów lub działań. Dane mogą też migrować wokół sieci w odpowiedzi na zmianę ich użycia lub wymagania dotyczące efektywności.



## 2. Przejroczystość fragmentacji

Użytkownicy nie muszą wiedzieć, w jaki sposób dane są podzielone. W naszym przykładzie zarządzania kadrami mamy do czynienia logicznie z jedną bazą danych, a fizycznie z czterema jej fragmentami. Dlatego np. kierownik uruchamiający zapytanie np. w Warszawie nie musi wiedzieć, że aby utworzyć łączną listę płac dla firmy, należy wykonać to zapytanie we wszystkich trzech miejscach – np. w Warszawie, Krakowie, Bydgoszczy i Gdańsku.

## 3. Przejroczystość replikacji

Użytkownicy nie muszą wiedzieć, w jaki sposób dane są powtarzane. W przykładowej bazie danych w każdym z czterech biur jest przechowywana kopia informacji o strukturze firmy. Kiedy zachodzi potrzeba okresowej aktualizacji tych danych, użytkownicy nie muszą być świadomi tego, że aktualizacja dotyczy każdego z czterech miejsc. Replikacja jest przydatna, gdyż efektywność rośnie, jeśli aplikacja może działać na lokalnych kopiach, a dostępność jest lepsza dopóty, dopóki co najmniej jedna kopia jest dostępna dla celów wyszukiwania danych [6].

## 4. Case study – Rozproszone systemy baz danych w planowaniu produkcji

W literaturze można znaleźć różne przykłady rozproszonych systemów baz danych w planowaniu produkcji. Gnonia i in. przedstawiają studium przypadku z branży motoryzacyjnej. Opisują oni problem wielkości partii i planowania (LSSP) wielostanowiskowego systemu produkcyjnego z ograniczeniami wydajności i niepewnym popytem na wiele produktów i okresów. LSSP rozwiązuje model hybrydowy wynikający z integracji modelu programowania liniowego o mieszanej liczbie całkowitej i modelu symulacyjnego. Podejście do modelowania hybrydowego jest stosowane w celu przetestowania lokalnej, jak i globalnej strategii produkcyjnej, w rozwiązaniu danego LSSP [7].

Inny przypadek przedstawiają Rupp i Ristic dotyczący planowania łańcuchów dostaw w produkcji półprzewodników [8]. Powszechnie przyjmuje się, że planowanie i kontrola produkcji w środowisku produkcji układów scalonych na zamówienie jest trudnym zadaniem, ponieważ musi być optymalna zarówno dla lokalnych jednostek produkcyjnych, jak i całej sieci łańcucha dostaw. Scentralizowane systemy MRP II, które działają w większości współczesnych przedsiębiorstwach produkcyjnych, nie są wystarczająco elastyczne, aby sprostać wymaganiom tego bardzo dynamicznego środowiska współpracy. Rupp i Ristic przedstawiają metodologię planowania rozproszonego dla łańcuchów dostaw produkcji półprzewodników.

Opracowany system opiera się na podejściu, które pozostawia tyle samo odpowiedzialności i wiedzy specjalistycznej do optymalizacji, jak to możliwe, lokalnym systemom planowania, podczas gdy globalny podmiot koordynujący zapewnia najlepszą wydajność i wydajność całego łańcucha dostaw.

Inny przykład przedstawia system składający się z dwóch zakładów produkcyjnych, fabryki 1 (F1) i fabryki 2 (F2), dla których należy uzyskać optymalny plan produkcji, z dwoma zapasami wyjściowymi i dwoma zewnętrznymi zakładami produkcyjnymi o nazwie Podwykonawca 1 i Podwykonawca 2 (Podwykonawca 1 przekazuje produkty końcowe do F1, a podwykonawca 2 do F2) [9]. Czas realizacji produkcji każdego zakładu jest równy jednemu okresowi (między fabrykami lub podwykonawcami). W podejściu zdecentralizowanym istnieją dwa zintegrowane problemy lokalnej optymalizacji od końca do początku. Mianowicie, najpierw należy zoptymalizować plan produkcji F2, a następnie F1. Z drugiej strony, w globalnej optymalizacji należy wziąć pod uwagę wszystkie cechy produkcji w F1 i F2 jednocześnie, a następnie należy zoptymalizować system globalnie.

W studium przypadku, przedstawionym w tym artykule, wyróżniono trzy przedsiębiorstwa produkcyjne, które są kontrolowane przez jeden zarząd. Wszystkie zakłady produkują elementy z tworzyw sztucznych. Każdy zakład posiada dział zakupów, produkcji i sprzedaży. Natomiast tylko jeden ma księgowość i finanse. Od czasu do czasu następuje wymiana danych dotyczących planowania produkcji oraz zakupów surowców. Co określony czas zarząd firmy zbiera informacje całościowe z produkcji tych trzech zakładów i optymalizuje ją pod względem zasobów magazynowych i ludzkich oraz produkcji. Ta optymalizacja jest niezwykle ważna ze względu na sezonowość produkowanych wyrobów. Zakłady pracują w układzie dwóch sezonów: w wysokim (od marca do sierpnia) i niskim (od września do lutego). Jeśli chodzi o produkcję w sezonie wysokim zmienia się kalendarz pracy w układzie trzech zmian przez siedem dni w tygodniu, natomiast w sezonie niskim zakłady przechodzą na trzy zmiany przez pięć dni w tygodniu. W sezonie wysokim musi istnieć w magazynach zapas bezpieczeństwa w postaci wyprodukowanych wyrobów w wymaganej liczbie. Stąd niezbędna jest ta optymalizacja. Na rys. 5 pokazano strukturę jednorodnych rozproszonych baz danych obejmujących trzy zakłady produkcyjne mieszczące się w trzech lokalizacjach. Bazy danych pracują w systemie Oracle pod systemem operacyjnym Windows Server 2012.

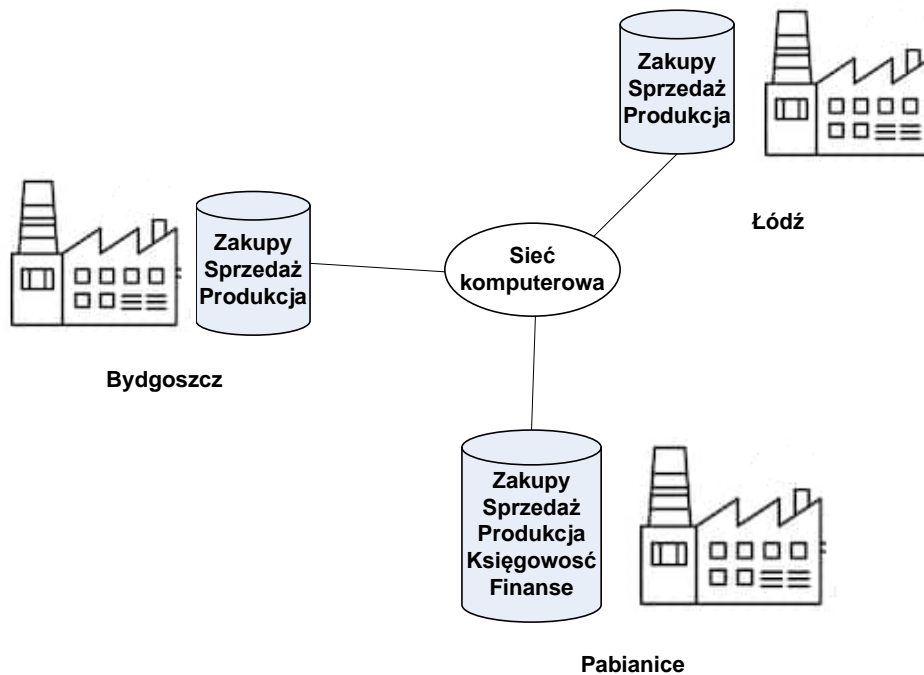
## 5. Podsumowanie

Rozproszone systemy baz danych dają ogromne możliwości światowym korporacjom rozproszonym na różnych kontynentach. Poszczególne oddziały firm jak również samodzielne firmy zrzeszone w korporacje mogą działać lokalnie optymalizując swoje działania lokalne jak również działają globalnie jako jeden organizm w sytuacjach gdy jest to wymagane.

Korporacje jak i firmy podejmują właściwe decyzje na poziomie lokalnym i globalnym co odpowiada różnorodności współczesnego świata.

W trakcie globalizacji wiele przedsiębiorstw zmienia swoje strategie i łączy je partnerstwo z dostawcami, podwykonawcami i klientami. To połączenie tworzy łańcuchy dostaw obejmujące kilka geograficznie rozmieszczonych zakładów produkcyjnych. Planowanie produkcji w łańcuchu dostaw jest skomplikowanym i trudnym zadaniem,

ponieważ musi być optymalne zarówno dla lokalnych jednostek produkcyjnych, jak i dla całej sieci łańcucha dostaw. W prawdziwym świecie operacyjnym, ze względów konkurencyjnych i / lub praktycznych, często każde przedsiębiorstwo woli optymalizować swój plan produkcji, nie zwracając uwagi na pozostałych członków łańcucha dostaw. Globalna optymalizacja tego rodzaju łańcucha dostaw jest trudna. Ponieważ zadanie planowania jest duże, cykle planowania mogą potrwać kilka godzin lub nawet dni i muszą być oparte na ustalonych czasach realizacji. Nie można brać pod uwagę ograniczonej dostępności zasobów i mocy lokalnych zakładów. Systemy te wytwarzają główne harmonogramy produkcji (MPS) i działają tylko wtedy, gdy centralna jednostka planowania stale otrzymuje informacje o bieżącym stanie produkcji rozproszonych lokalizacji.



**Rysunek. 5** Jednorodny rozproszony system baz danych wybranego przedsiębiorstwa produkcyjnego

## Literatura

1. Garcia Molina H., Ullman J.D., Widom J. Systemy baz danych. Pełny wykład. WNT, Warszawa 2006.
2. Andrzejewski W., Królikowski Z., Morzy T. Bazy danych i systemy informatyczne oraz ich wpływ na rozwój informatyki w Polsce. In: Polskie i światowe osiągnięcia nauki: Nauki techniczne, 2010, 345–388.
3. Rojek I. Bazy danych i bazy wiedzy w zarządzaniu wiedzą technologiczną przedsiębiorstwa, red. S. Kozielski, B. Małysiak, P. Kasprowski, D. Mrozek, Bazy Danych – Modele, Technologie, Narzędzia, Wydawnictwo Komunikacji i Łączności, s. 257-264, Warszawa 2005.
4. Wrembel R., Bębel B. Oracle Projektowanie rozproszonych baz danych. Helion, Warszawa 2003.
5. Brzeziński M. Sterowanie produkcją. Wyd. Politechniki Lubelskiej, Lublin 2009.
6. Mosorow W. Systemy rozproszone, <http://mosorow.kis.p.lodz.pl/pl/sr/sr.pdf> - data pobrania 31.10.2019
7. Gnonia M.G., Iavagnilio R., Mossa G., Mummolo G., Di Leva A. Production planning of a multi-site manufacturing system by hybrid modelling: A case study from the automotive industry. Int. J. Production Economics, 2003; 85:251-262.
8. Rupp T.M., Ristic M. Fine planning for supply chains in semiconductor manufacture, Journal of Materials Processing Technology, 2000; 107:390-397.
9. Saharidis G., Dallery Y., Karaesmen F. Centralized versus decentralized production planning, [http://home.ku.edu.tr/~fkaraesmen/pdffpapers/SDK\\_revised\\_05.pdf](http://home.ku.edu.tr/~fkaraesmen/pdffpapers/SDK_revised_05.pdf) - data pobrania 31.10.2019

# GRUPY JEDNOLITE W METODZIE REPREZENTACJI BINARNYCH

**Janusz Łukowski**

Uniwersytet Kazimierza Wielkiego  
Instytut Mechaniki i Informatyki Stosowanej  
ul. Kopernika 1 p.207a, 85-074 Bydgoszcz  
e-mail: januszl@ukw.edu.pl

**Streszczenie:** W artykule przedstawiono definicję grupy jednolitej zdefiniowanej w metodzie reprezentacji binarnych. Możliwość identyfikacji grupy jednolitej umożliwia konstrukcję uproszczonego zapisu logicznego w całościowym rozpatrywaniu układu kombinacji wielowejsściowego układu kombinacyjnego. Metoda reprezentacji binarnych stanowi alternatywny sposób konstrukcji uproszczonego zapisu funkcji wyjściowej układu kombinacyjnego w odniesieniu do metody przekształceń formalnych, metody tablic Karnaugh'a czy metody Quine'a–McCluskeya.

**Słowa kluczowe:** Tabela prawdy układu kombinacyjnego, grupa jednolita, metoda reprezentacji binarnych

## *Uniform group in the binary representation method*

**Abstract:** The article presents definitions of uniform group defined in the binary representation method. The ability to identify uniform group enables the construction of a simplified logical description in the total consideration of the combinations of a multi-input combination system. The binary representation method is an alternative way of constructing a simplified description of the output function of a combinational system in relation to the method of formal transformations, the Karnaugh table or the Quine-McCluskey methods.

**Key words:** Truth table of combinational system, uniform group, associated group, method of binary representations

## 1. Wprowadzenie

Metoda reprezentacji binarnych stanowi alternatywny sposób konstrukcji uproszczonego zapisu funkcji logicznej układu kombinacyjnego w odniesieniu do metody przekształceń formalnych [1,2,4,8], metody tablic Karnaugh'a [7], metody Quine'a–McCluskeya [6] lub metody bezpośredniego przeszukiwania [4]. Istotą metody reprezentacji binarnych jest:

- podział kombinacji układu na zestawy po 4 kombinacje każdy;
- identyfikacja zgodności tabeli prawdy podstawowej funkcji boolowskiej lub układu kombinacji jednej ze zmiennych wejściowych z postacią funkcji wyjściowej w obrębie zestawu;
- w przypadku zgodności uwzględnienie pozostałych zmiennych wejściowych w zapisie logicznym funkcji, przy założeniu stanu wysokiego funkcji wyjściowej dla każdej kombinacji w obrębie zestawu.

Zestawy kombinacji mogą tworzyć grupy jednolite pozwalające na całościową analizę

wszystkich kombinacji układu kombinacyjnego i na ich podstawie konstrukcję uproszczonego zapisu funkcji wyjściowej układu kombinacyjnego.

## 2. Grupa jednolita

Grupą jednolitą nazywamy układ zestawów układu kombinacyjnego charakteryzujący się identyczną postacią kombinacji stanów wyjściowych dla każdego zestawu wchodzącego w skład grupy. Konfiguracja kombinacji stanów wyjściowych zestawów grupy jednolitej może być:

- zgodna z postacią tabeli stanów funkcji boolowskiej;
- zgodna z układem kombinacji jednego z wejść układu kombinacyjnego;
- niezgodna z postacią tabeli stanów funkcji boolowskiej lub układem kombinacji jednego z wejść układu w przypadku jednego, dwóch lub trzech (rozpatrywanych w układzie  $2 + 1$  lub  $1 + 2$ ) identycznych stanów wyjściowych dla zestawów.

W Przykładzie 1 przedstawiono przykładowe postacie grup jednolitych dla układu kombinacyjnego trójwejściowego.

Przykład 1

x <sub>2</sub>	x <sub>1</sub>	x <sub>0</sub>	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>
0	0	0	0	0	1	0
0	0	1	1	0	1	1
0	1	0	1	1	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	1	0	0	1
1	1	0	1	1	1	0
1	1	1	0	1	1	0

W przypadku, gdy grupę jednolitą tworzy funkcja logiczna dwóch zmiennych wejściowych lub gdy kombinacje stanów wyjściowych są identyczne z kombinacjami jednej ze zmiennych wejściowych, to celem uwzględnienia pozostałych zmiennych wejściowych w zapisie logicznym przyjmujemy stan wysoki (jedynka logiczna) dla wszystkich kombinacji stanów wyjściowych zestawów tworzących grupę jednolitą.

Układ trójwejściowy posiada osiem kombinacji tworzących dwa zestawy. W przypadku stanów wyjścia y<sub>0</sub> mamy do czynienia z dwoma zestawami, w których kombinacje stanów wejść odpowiadają funkcji logicznej XOR dla wejść x<sub>1</sub>x<sub>0</sub>.

Dla stanów wyjścia y<sub>1</sub> mamy do czynienia z dwoma zestawami, w których kombinacje stanów wejść odpowiadają kombinacjom zmiennej wejściowej x<sub>1</sub>.

Zmiana stanu wejść układu kombinacyjnego wielowejściowego jest zgodna z zapisem: tj. dla wejścia x<sub>0</sub> zmiana stanu następuje co 1 krok, dla wejścia x<sub>1</sub> zmiana stanu wejścia następuje co 2, itd.

Przykład 1a

x <sub>2</sub>	x <sub>1</sub>	x	y	y
		0	0	1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

W przypadku wyjścia y<sub>2</sub> (Przykład 1) mamy do czynienia z podobną sytuacją jak w Przykładzie 1a dla funkcji XNOR, wówczas mamy: . Dla stanów wyjścia y<sub>3</sub> mamy do czynienia z dwoma zestawami posiadającymi identyczną postać stanów wyjściowych lecz nie

posiadającą postaci funkcji boolowskiej i stanu jednego z wejść układu kombinacyjnego. Oba zestawy rozpatrujemy łącznie. W łącznym rozpatrywaniu stanów wyjściowych dla zestawów tworzących grupę jednolitą porównujemy ze sobą stany zmiennych wejściowych. W przypadku, gdy wartości zmiennych wejściowych dla rozpatrywanych stanów wyjściowych są różne, to dana zmienna wejściowa nie występuje w zapisie logicznym funkcji.

### 3. Dostosowanie stanu wyjść układu kombinacyjnego do postaci grupy jednolitej

W przypadku, gdy chcemy utworzyć grupę jednolitą dla układu kombinacyjnego, w którym jedna lub kilka kombinacji nie odpowiada założeniom grupy jednolitej możliwa jest korekta stanu wyjściowego układu uwzględniająca zastosowaną postać funkcji logicznej lub kombinację jednego z wejść układu kombinacyjnego.

Przykład 2

x <sub>2</sub>	x	x	y	
	1	0		
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Utworzenie grupy jednolitej wykorzystującej funkcję XOR, dla przypadku przedstawionego w Przykładzie 2, wymaga wyeliminowania 1 kombinacji wejść układu (zaznaczonej na szaro). W tym celu należy uwzględnić w zapisie dysjunkcyjnym funkcji logicznej y stan niski w kombinacji w postaci koniunkcyjnej.

Przykład 3

x <sub>2</sub>	x	x	y	
	1	0		
0	0	0	1	1
0	0	1	0	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	0

Utworzenie grupy jednolitej wykorzystującej funkcję NOR, dla przypadku przedstawionego w Przykładzie 3, wymaga uwzględnienia 1 kombinacji wejść układu (zaznaczonej na szaro). W tym celu należy uwzględnić w zapisie dysjunkcyjnym funkcji logicznej y stan wysoki 1 kombinacji w postaci dysjunkcyjnej.

#### 4. Podsumowanie i wnioski

Możliwość konstrukcji grup jednolitych układu kombinacyjnego pozwala na całościową analizę wszystkich kombinacji i na ich podstawie konstrukcję uproszczonego zapisu funkcji wyjściowej układu kombinacyjnego. Przedstawiona technika dostosowania stanu wyjść układu kombinacyjnego do postaci grupy jednolitej pozwala konstrukcję zapisu logicznego funkcji wyjściowej układu kombinacyjnego wykorzystującej zapis logiczny funkcji logicznej dwuargumentowej lub zmiennej wejściowej.

#### Literatura

1. Kalisz J.: Podstawy elektroniki cyfrowej, WKŁ, 1993
2. Majewski W.: Układy logiczne, WNT, 1993
3. Mano M.M.: Computer engineering: hardware design, Prentice-Hall, 1988
4. Traczyk W.: Układy cyfrowe. Podstawy teoretyczne i metody syntezy, WNT, 1986
5. Bromirski J.: Teoria automatów WNT, 1969
6. McCluskey E.: Logic design principles, Prentice-Hall, 1986
7. Kamionka-Mikuła H., Małysiak H., Pochopień B.: Synteza i analiza układów cyfrowych. Gliwice: Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, 2009
8. Adamowicz A., Zbierski P.: Logic of mathematics. A modern course of classical logic. Nowy Jork: A Wiley-Interscience Publication. John Wiley & Sons, Inc., 1997, seria: Pure and Applied Mathematics.