

Wojciech PIETROWSKI\*  
Łukasz KRYGIER\*  
Wojciech MACIEJEWSKI\*  
Konrad GÓRNY\*

## **SYSTEM APLIKACJI MOBILNEJ DO BEZPRZEWODOWEGO MONITOROWANIA URZĄDZEŃ**

W artykule zaprezentowano autorską aplikację mobilną wraz z układem pomiarowym składającym się z czujnika temperatury oraz systemu do gromadzenia i archiwizowania danych. Opisano wykorzystaną technologię transmisji danych, cechy i możliwości platformy Arduino oraz środowisko programistyczne Xamarin. Zaprezentowano schemat działania kompletnego układu wraz z serwerem oraz bazą danych, w której gromadzone są wyniki pomiarów. Przedstawiono interfejs aplikacji mobilnej z jej dostępnymi funkcjami.

SŁOWA KLUCZOWE: układ czujników, Wi-Fi, Arduino, system Android, Xamarin, aplikacja mobilna

### **1. WPROWADZENIE**

Rosnąca popularność rozwiązań mobilnych, łatwość dostępu do sieci Internet i technologii bezprzewodowej wymiany danych a także ciągły postęp dziedziny związanej z diagnostyką urządzeń elektrycznych wpływają na rozwój bezprzewodowych systemów diagnostyczno-kontrolnych. Niewątpliwie możliwość kontroli pomiarów związanych z parametrami pracy maszyny z dowolnego miejsca przy użyciu na przykład ekranu telefonu komórkowego, wpływa na sprawniejszy monitoring, a także zwiększa szansę wychwycenia nieprawidłowości w działaniu maszyny. Aktualnie a rynku obecność rozwiązań podobnych do przedstawionego w artykule jest znikoma. Artykuł przedstawia kompletny system diagnostyczny, umożliwiający rozwój platformy w przyszłości. W artykule skupiono się na komunikacji urządzeń, a także odpowiednim archiwizowaniu wyników. Jako sposób komunikacji wybrano protokół Wi-Fi i przyjęto założenie, że zarówno badane urządzenie jak i telefon komórkowy, na którym wyświetlano wyniki, znajdują się w zasięgu tej samej sieci lokalnej, dokonano wyboru drogi komunikacji między urządzeniami.

---

\* Politechnika Poznańska.

Skorzystano z mikrokontrolera Arduino, a także zewnętrznego modułu Wi-Fi. Wyniki pomiarów w ustalonych interwałach czasowych wysyłane są poprzez serwer WWW do bazy danych i tam gromadzone. Następnie, po uruchomieniu aplikacji na telefonie, rekordy pobierane są do pamięci smartfona. Z uwagi na fakt, że serwery dysponują większą mocą obliczeniową w odniesieniu do telefonów komórkowych, wszystkie operacje wykonywane są po stronie serwera. Aplikacja służy do wizualizacji wyników w poszczególnych zakładkach wraz z datą i godziną pomiaru oraz średnią dla każdego dnia, a także przedstawia zależności zmiany badanego parametru pracy w formie wykresu. Aplikację wyposażono także w ustawienia użytkownika, które pozwalają na zmianę podstawowych aspektów jej funkcjonowania takich jak liczba pobieranych wyników, czy ustawienie ostrzeżenia o przekroczeniu wartości krytycznej dla konkretnego pomiaru.

## 2. PRZETWARZANIE I WYSYŁANIE DANYCH

Platforma Arduino ze względu na swoją cenę, dostępność a także system dystrybucji w postaci Open Hardware stanowi efektywne narzędzie które może być wykorzystane w celach diagnostyki urządzeń elektrycznych. Rosnąca popularność platformy Arduino przyczyniła się również do innych jej zastosowań do których należą na przykład:

- proste gry,
- roboty podążające wzdłuż linii (tzw. line-followery),
- sterowanie oświetleniem,
- roboty samodzielnie odzyskujące równowagę,
- inteligentne sterowanie w budynkach.

W zależności od potrzeb wyróżniamy wiele różnych wersji Arduino. Większość z nich jest oparta na 8-bitowym mikrokontrolerze z rodziny AVR firmy Atmel. Pod koniec 2012 roku na rynku pojawił się nowy model Arduino – DUE, które jako pierwsze wyposażone zostało w 32-bitowy procesor z rdzeniem ARM [1].

Open Hardware'owa dystrybucja platformy pozwala również na tworzenie własnych mikrokontrolerów w oparciu o dokumentację urządzeń. Co więcej, każdy może stworzyć swoją wersję do własnego użytku. Największą popularnością cieszy się Arduino Uno z mikrokontrolerem Atmega328.

Mikrokontroler ATMEGA328 używany jest przeważnie w prostych autonomicznych systemach o małej mocy. Taki układ można programować za pomocą dwóch języków programowania: Assembler oraz język C. Programowanie samego mikrokontrolera ATMEGA328 wymaga wiedzy z zakresu budowy i zasady działania układu. Zastosowanie platformy Arduino ze względu na swoją architekturę pozwala na eliminację tego problemu.

## 2.1. Moduły Arduino

Platforma Arduino ma możliwość rozbudowywania funkcjonalności poprzez moduły. Jest to dobre rozwiązanie, lecz w przypadkach, gdzie używana jest duża ilość wejść/wyjść warto zapoznać się z tzw. shieldami. Są to nakładki, które dzięki budowie podobnej do Arduino, służą jako moduły bez konieczności niwelowania ilości wolnych pinów. Dodatkową zaletą jest możliwość podłączenia w ten sposób nie tylko jednej, ale i kilku takich nakładek co znacznie zmniejsza ilość przewodów potrzebnych do podłączenia układu, a także ujednolica konstrukcję.

Do bezprzewodowej komunikacji między urządzeniami wykorzystywana jest sieć Wi-Fi. W prezentowanym systemie wykorzystano nakładkę SparkFun ESP8266 Shield, która umożliwi bezprzewodową transmisję danych w Arduino.

## 2.2. Czujnik temperatury DS18B20

Do pomiaru temperatury w układzie służy wodoodporna sonda. Posiada ona wbudowany cyfrowy czujnik o zakresie pomiarowym od  $-55^{\circ}\text{C}$  do  $125^{\circ}\text{C}$ . Zasilana jest napięciem od 3 V do 5,5 V. Kompatybilności czujnika z platformą Arduino oraz dostępność bibliotek umożliwia bezawaryjną pracę urządzenia. Połączenie czujnika realizowane jest poprzez 3 przewody: zasilania VCC, masy GND oraz 1-wire do przesyłania danych.

## 3. TECHNOLOGIA BEZPRZEWODOWEJ TRANSMISJI DANYCH WI-FI

Standardy typu Wi-Fi opierają się na bezprzewodowej transmisji wykorzystującej zakres częstotliwości 2,4 GHz oraz 5 GHz. Na dokładną stosowaną częstotliwość składa się liczba używanych kanałów w transmisji, ta z kolei różni się w zależności od kraju (w Polsce używa się 13 kanałów, podczas gdy np. w Japonii używa się ich 14). Nie mniej jednak, celem zachowania ogólnoświatowego, jednolitego standardu, numeracja kanałów pozostaje niezmienną, tak więc kanał numer 3 odpowiada kanałowi nr 3 o tej samej częstotliwości niezależnie od kraju [2].

Prędkości przesyłania danych, jakie oferuje Wi-Fi różnią się w zależności od użytego standardu. Dla najnowszych standardów osiągnąć można prędkości do 600 Mb/s [2].

Wadą systemów typu Wi-Fi jest zasięg działania sieci, z reguły nie przekracza on 500 metrów, co oznacza, iż w kraju wielkości Polski, sieć Wi-Fi może być na ten moment w kontekście mobilności jedynie uzupełnieniem takich standardów jak np. LTE (*ang. Long Term Evolution*).

Sieci typu Wi-Fi znajdują zastosowanie głównie w domach i firmach, celem zbudowania sieci z dostępem do Internetu pozbawionej okablowania. Dodatkowo w budynkach użyteczności publicznej spotkać można także tak zwane HotSpoty, to jest miejsca z dostępem do sieci Wi-Fi, do której połączyć możemy urządzenie. Jeden z największych HotSpotów, używając 650 punktów dostępu osiąga zasięg 140 hektarów [3].

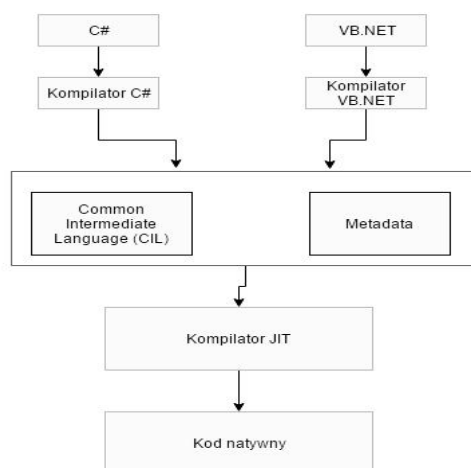
W projekcie mobilnej aplikacji do monitorowania urządzeń zdecydowano się na wybór standardu Wi-Fi celem transmisji danych z czujników do serwera bazy danych. Argumenty to dostępność sieci Wi-Fi praktycznie w każdym zakładzie produkcyjnym, łatwość ustanowienia połączenia lokalnego oraz niewielkie koszty przedsięwzięcia.

#### **4. ZAPISYWANIE I GROMADZENIE DANYCH**

Prezentowany system w celu stworzenia lokalnego serwera korzysta z rozwiązania firmy Microsoft. IIS (Internet Information Services) jest zbiorem usług oferowanych dla systemów z grupy Microsoft Windows oferujących funkcje serwera FTP, HTTP, HTTPS czy SMTP. Według statystyk jest drugim pod względem popularności serwerem internetowym na świecie. Częściej spotykany jest tylko serwer Apache. Serwery są specjalistycznymi komputerami z oprogramowaniem, które umożliwia przyjmowanie żądań od klienta i zwracanie odpowiedzi [4].

##### **4.1. Platforma Microsoft.NET**

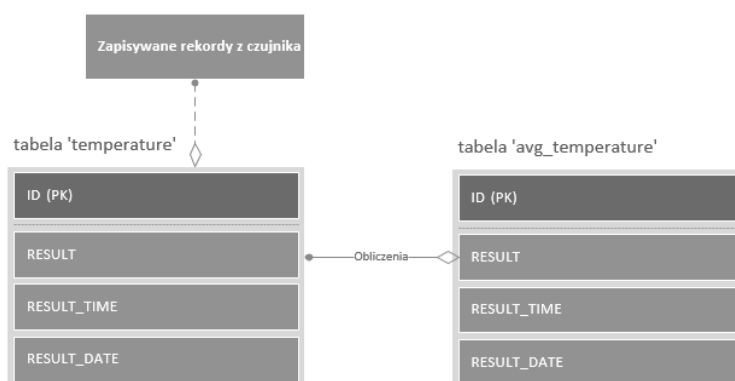
W celu oprogramowania serwera skorzystano z platformy .NET. Jest to technologia wprowadzona na rynek przez firmę Microsoft. Rozwiązanie to kreuje środowisko uruchomieniowe dla aplikacji oraz zbiór bibliotek, które pozwalają na wprowadzanie funkcjonalności dla aplikacji. Firma Microsoft, tworząc platformę korzystała ze znanych z języka Java rozwiązań, opierających się na tym, że kod źródłowy kompilowany jest do uniwersalnego kodu, który nazywa się kodem pośrednim (CIL). Należy nadmienić, że kod pośredni jest wykonywany przez maszyny wirtualne, a nie procesor, sprawia to, że jest on niezależny od architektury sprzętowej. Istotna jest także tak zwana kompilacja w locie (ang. *just in time – JIT*), która działa w taki sposób, że metoda danej klasy kompilowana jest w momencie jej pierwszego wywołania bezpośrednio do kodu maszynowego, a kolejne wywołania tej metody odwołują się już do skompilowanego kodu. Kompilowane jest niezbędne minimum, wymagane do uruchomienia aplikacji, reszta kodu dopiero, gdy zajdzie taka potrzeba. Wpływa to znacząco na wydajność aplikacji. Na rys. 1 przedstawiono schemat kompilacji kodu w .NET [5].



Rys. 1. Schemat kompilacji kodu w .NET

#### 4.2. Baza danych

Stworzona na potrzeby aplikacji baza danych składa się z dwóch tabel dla każdego z czujników. W pierwszej z nich zapisywane są kolejne wyniki wraz z ich datą i godziną zapisu. W drugiej z kolei zapisywane są średnie wartości wyników dla poszczególnych dni, w których wykonywane są pomiary. Na rys. 2 przedstawiono schemat współdzielenia tabel baz danych.



Rys. 2. Schemat współdzielenia tabel bazy danych

Trzecią tabelą jest związana z ustawieniami. Przechowywane są w niej wszystkie parametry ustawień zapisane przez użytkownika.

## 5. OPIS ZAPROJEKTOWANEGO SYSTEMU

Sonda pomiarowa z wbudowanym czujnikiem temperatury podłączona do pinu cyfrowego Arduino przesyła informacje o bieżącej temperaturze. Dzięki mikrokontrolerowi ATMEGA328 pobrane dane są przetwarzane, a następnie wysyłane poprzez moduł Wi-Fi SparkFun ESP8266 na serwer lokalny znajdujący się w tej samej sieci. Metodą GET protokołu HTTP dane pobierane są przez stronę internetową umieszczoną na tym serwerze, gdzie następnie gromadzone są w bazie danych. Aplikacja mobilna zainstalowana na urządzeniu z systemem Android, dzięki odpowiednim zapytaniom do bazy danych pobiera te wyniki, a następnie wizualizuje je w odpowiednich zakładkach.

## 6. BUDOWA I FUNKCJE APLIKACJI

Xamarin to platforma kupiona przez firmę Microsoft, początkowo rozwijana w projektach Mono czy Mono For Android. Zapewnia ona możliwość pisania natywnych aplikacji na system Android, iOS, a także mobilne wersje Windowsa. Jest częścią bardzo popularnego IDE Microsoftu – Visual Studio. Środowisko Xamarin umożliwia programowanie w języku C#. Do zalet aplikacji zaliczyć można dostęp do chmury, w której to poprzez emulację różnych urządzeń, testować można napisany uprzednio kod. Środowisko Xamarin umożliwia także korzystanie niemal ze wszystkich zalet języka C# i platformy .NET.

### 6.1. Funkcje aplikacji

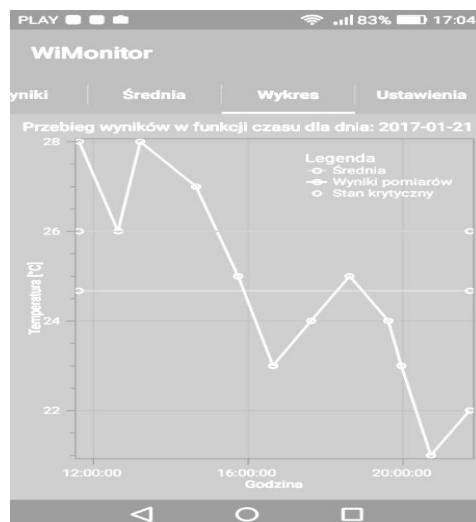
Prezentowana aplikacja posiada szereg funkcji służących do wizualizacji otrzymanych wyników pomiarów. Całość pogrupowana jest w odpowiednich zakładkach. Pierwszą z nich jest zakładka „Wyniki”, w której to wyświetlane są kolejne wyniki w ustalonej przez użytkownika ilości. Oprócz samej temperatury aplikacja umożliwia wyświetlenie informacji o dacie oraz godzinie wykonania pomiaru. Dodatkowo poprzez zaimplementowaną funkcję możliwe jest ustalenie wartości granicznej temperatury. Kiedy zostanie ona przekroczona, wynik wyświetla się na czerwonym tle, natomiast wynik zbliżony do tej wartości zostaje oznaczony kolorem pomarańczowym. Aktualizacja pobranych danych jest możliwa dzięki gestowi „SwipeToRefresh”, czyli przeciągnięcia palca w dół ekranu.

Na rysunku 3 przedstawiono wyniki działania opisanej powyżej funkcji. Lewa strona rysunku prezentuje wyniki przy ustalonej temperaturze granicznej 26°C, natomiast po stronie prawej temperatura graniczna wynosi 24°C. Kolejna z zakładek dotyczy wartości średniej. W zakładce tej wyświetlana jest średnia wartość temperatury dla kolejnych dni pomiarów a zaimplementowana wyżej funkcja informuje o przekroczeniu wartości granicznej w danym dniu.



Rys. 3. Oznaczone wartości graniczne

Trzecia zakładka przedstawia wykres zależności temperatury od czasu. Wyświetlane są wyniki z obecnego dnia pomiarów. Co więcej zaznaczona jest także średnia oraz wspomniana wcześniej linia wartości granicznej. Możliwe jest również przybliżanie, oddalanie oraz przesuwanie wykresu poprzez gest rozsuwania/zsuwania palców. Wykres aktualizuje się w momencie odświeżenia zakładki „Wyniki”. Rys. 4 przedstawia przykładowe wyniki działania programu.



Rys. 4. Widok temperatury

Przedstawiona na rys 5 zakładka „Ustawienia” pozwala na konfigurację takich aspektów działania aplikacji jak liczba wyświetlanych rekordów w zakładce „Wyniki”, ustalenie wartości granicznej temperatury czy zmiana motywu kolorystycznego.



Rys. 5. Zakładka ustawień

## 7. PODSUMOWANIE

Testy aplikacji na różnych urządzeniach z systemem nie wykazało błędów w jej funkcjonowaniu. Komunikacja między urządzeniami działa prawidłowo. Dane z czujników są pobierane i wyświetlane w interfejsie graficznym aplikacji. Ponadto zrealizowano cel związany z umożliwieniem rozbudowy systemu. Aplikacja pozwala na zmianę takich parametrów jak: typ serwera na dedykowany czy też dodanie obsługi kont użytkowników. Ważnym aspektem jest także możliwość dodania innych typów czujników. Dzięki możliwości rozwoju aplikacji może być ona w przyszłości wykorzystana w bardziej skomplikowanych systemach diagnostycznych.

## LITERATURA

- [1] Evans M., Noble J., Hochenbaum J.: Arduino w akcji, Helion, 2014.
- [2] Makowski D., Technologie komunikacji bezprzewodowej. Wprowadzenie do przedmiotu, <http://neo.dmcs.pl/>
- [3] <http://muniwireless.com/2003/06/20/university-of-twente-largest-european-hotspot/>
- [4] <https://news.netcraft.com/archives/2016/06/22/june-2016-web-server-survey.html>
- [5] <http://foreverframe.pl/just-in-time-czyli-kompilacja-w-net/>



**SYSTEM OF THE MOBILE APPLICATION  
TO THE WIRELESS MONITORING OF DEVICES**

The article presents mobile application with measuring system containing temperature sensor and data collection and archiving system including wireless data transfer technology, Arduino and Xamarin IDE. The scheme of sending and receiving data using local server and data base has been presented either. Also mobile application interface with available functionalities has been described.

*(Received: 20. 02. 2017, revised: 28. 02. 2017)*