

Patryk PACZKOWSKI\*\*  
Karol JÓZEFOWICZ\*\*  
Andrzej RYBARCZYK\*

## **STEROWANIE INTELIGENTNYM DOMEM PRZY UŻYCIU MOBILNEJ APLIKACJI WYKORZYSTUJĄCEJ MIKROKONTROLER ARDUINO UNO V3**

W niniejszej pracy został przedstawiony system sterowania inteligentnym domem z poziomu aplikacji mobilnej dla systemu Android wykorzystując przy tym platformę programistyczną Arduino UNO v3. W celu zobrazowania wyników działającego systemu została również wykonana w pełni funkcjonalna makieta. Zastosowanie Arduino, które charakteryzuje się dużymi możliwościami modyfikacji, umożliwia w praktyczny, a zarazem ciekawy sposób przedstawienia zagadnienia na rzeczywistym przykładzie modelu inteligentnego domu.

SŁOWA KLUCZOWE: Arduino, Android, inteligentny dom

### **1. WPROWADZENIE**

Projekt systemu sterowania inteligentnym domem za pomocą urządzenia z systemem Android ma na celu przedstawienie możliwości niedrogiego, a zarazem dysponującego dużymi możliwościami swobodnej rozbudowy o nowe funkcje, środowiska Arduino. Dla pełnego przedstawienia funkcjonalności systemu została również zbudowana makieta symulująca działanie inteligentnego domu. Bezprzewodowa komunikacja pomiędzy urządzeniami mobilnymi, a Arduino UNO odbywa się za pomocą technologii Bluetooth. Projekt obejmuje takie stany sterowania jak kontrola oświetlenia, sterowanie drzwiami, odczyt temperatury, symulacja ogrzewania oraz zapalenie się światła po wykryciu ruchu w pomieszczeniach domu. Urządzeniami wykonawczymi realizującymi te stany sterowania są m.in.: diody LED, mikro serwomechanizmy, czujniki ruchu oraz czujnik temperatury. Arduino wraz z urządzeniami wykonawczymi zasilane jest baterią o napięciu 9 V (można również użyć zasilacza impulsowego 12 V). Do poprawnego funkcjonowania całego systemu zostało przygotowane oprogramowanie zarówno dla systemu Android, jak i dla płytki Arduino

---

\* Politechnika Poznańska.

\*\* Państwowa Wyższa Szkoła Zawodowa w Lesznie.

UNO v3 (wyposażonej w mikrokontroler Atmega328). Oprogramowanie dla Arduino zostało napisane w języku C.

## 2. OPROGRAMOWANIE STERUJĄCE

### 1.2. Oprogramowanie dla systemu Android

Zaprezentowana aplikacja na platformę Android potwierdza jej zalety, do których należy przede wszystkim zapewnienie w dużym stopniu możliwości uniknięcia niebezpiecznych stanów przy tworzeniu nowych aplikacji. Jest to oczywiście wynikiem tego, że urządzenia oparte na platformie Android działają pod kontrolą systemu operacyjnego Linux, którego zaletą jest właśnie bezpieczeństwo tworzenia aplikacji. Wszystkie aplikacje uruchamiane na platformie Android działają w oparciu o własną wirtualną maszynę DALVIK, która bazuje na wirtualnej maszynie Javy, ale została ona dostosowana do urządzeń przenośnych. Dodatkową zaletą takiego rozwiązania jest to, że wirtualna maszyna DALVIK zajmuje bardzo mało miejsca w pamięci oraz na urządzeniu można uruchomić jednocześnie wiele jej egzemplarzy.

Kiedy w telefonie opartym na platformie Android jest instalowana aplikacja, to system tworzy jej indywidualny profil użytkownika, który posiada swoje prywatne pliki, własny identyfikator oraz bezpieczne środowisko, w którym jest wykonywana. Takie rozwiązanie zmniejsza ryzyko potencjalnej awarii całego telefonu poprzez niepoprawnie działającą aplikację oraz wyklucza jego uszkodzenie. Usługi systemowe platformy Android obsługiwane są przez jądro Linuksa, którego podstawową funkcją jest pośredniczenie pomiędzy podzespołami telefonu, a programowalnym stosem Androida, czynność ta przypomina działanie warstwy abstrakcji sprzętowej (ang. *hardware abstraction layer* – HAL).

Jądro systemu realizuje takie funkcje, jak: wymuszenie zasad bezpieczeństwa aplikacji, zarządzanie pamięcią na niskim poziomie, zarządzanie procesami i wątkami, obsługa stosu sieciowego, dostęp do sterowników podzespołów.

Zadaniem przygotowanego projektu oprogramowania dla urządzeń mobilnych z systemem Android było sterowanie funkcjonalnościami inteligentnego domu w prosty i intuicyjny sposób. Najistotniejszym elementem programu jest wykorzystanie połączenia przez Bluetooth, a następnie wysyłanie odpowiednich komend do modułu Bluetooth urządzenia odbiorczego, czyli Arduino UNO. Polecenia wysyłane są w zależności od efektu, który chcemy wywołać w inteligentnym domu: np. gdy chcemy zapalić światło w kuchni to po naciśnięciu przycisku z tą funkcjonalnością Android wysyła do Arduino zmienną przechowującą ciąg znaków (typu string) o wartości „TOK”. Opisany przykład odzwierciedlony został we fragmencie kodu, który zawarty jest w tabeli 1.



Rys. 1. Interfejs aplikacji mobilnej do sterowania inteligentnym domem

Tabela 1. Fragment kodu oprogramowania odpowiedzialnego za sterowanie oświetleniem w kuchni

```
public void włączenieSwiatlaKuchnia(View view) {
    if (btnKuchnia.isChecked()) {
        if (btSocket != null) {
            try {
                btSocket.getOutputStream().write("TOK".toString().getBytes());
            } catch (IOException e) {
                msg("Problem z włączeniem światła w Kuchni");
            }
        }
    } else {
        if (btSocket != null) {
            try {
                btSocket.getOutputStream().write("TOFK".toString().getBytes());
            } catch (IOException e) {
                msg("Problem z wyłączeniem światła w kuchni");
            }
        }
    }
}
```

## 2.2. Oprogramowanie dla platformy Arduino

Oprogramowanie napisane dla platformy Arduino UNO ma za zadanie odebranie komend wysyłanych z aplikacji mobilnej poprzez Bluetooth i następnie jej zweryfikowanie oraz uruchomienie odpowiedniego stanu sterowania. Na początku działania jest ustalany port, który będzie gotowy do odbioru komend z urządzenia mobilnego przez Bluetooth, a potem następuje zainicjowanie modułu do działania. Dla przykładu, polecenie zaświecenia się diody, odpowiedzialnej za symulację oświetlenia w kuchni wymaga, aby Arduino odebrało polecenie „TOK” i zrealizowało funkcję odpowiedzialną za ten element. W tabeli 2 pokazano odpowiedni fragment kodu tego polecenia.

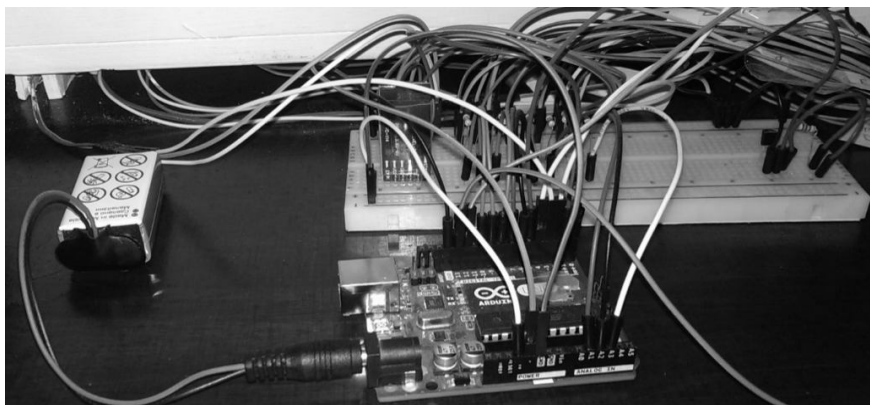
Tabela 2. Fragment kodu oprogramowania dla Arduino odpowiedzialny za sterowanie oświetleniem kuchni

<pre>if (string == "TOK") {   onLedKuchnia();   ledon = true; }</pre>	<pre>void onLedKuchnia() {   analogWrite(kuchniaLed, 255);   delay(10); }</pre>
---	---

## 3. WERYFIKACJA PROJEKTU



Rys. 2. Makieta inteligentnego domu



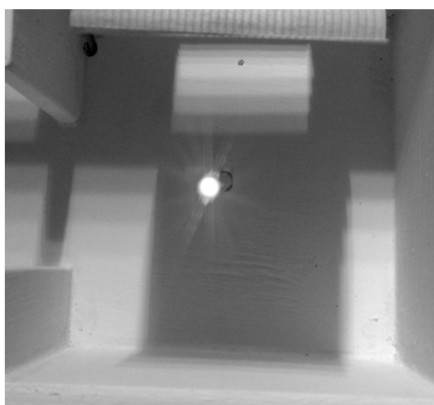
Rys. 3. Gotowe do pracy Arduino połączone z urządzeniami wykonawczymi

## 4. WERYFIKACJA STANÓW STEROWANIA

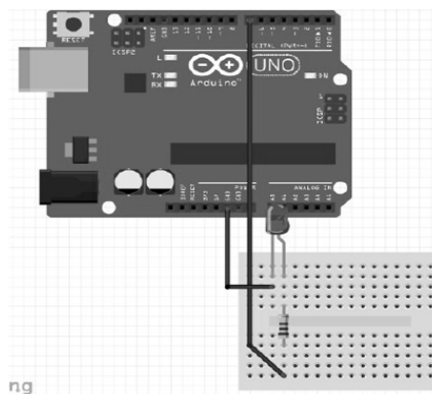
### 4.1. Oświetlenie – diody LED

Na rys. 4 przedstawiono działanie diody LED, która jest odpowiedzialna za symulację oświetlenia w salonie. W projekcie zostały zastosowane diody LED, do których zasilania potrzebny jest prąd o wartości 25mA oraz napięciu 2V.

Na rys. 4 przedstawiono działanie diody LED, która jest odpowiedzialna za symulację oświetlenia w salonie.



Rys. 4. Weryfikacja stanu działania oświetlenia w salonie



Rys. 5. Schemat podłączenia diody LED do Arduino UNO v3

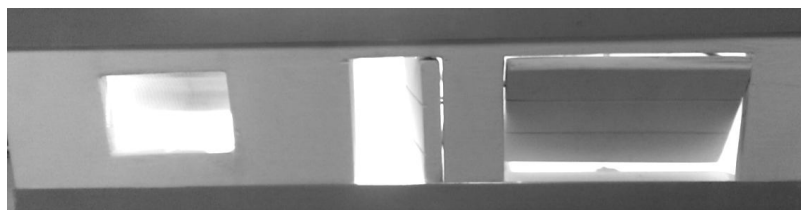
## 4.2. Sterowanie drzwiami – serwomechanizmy

Serwomechanizm można określić, jako mechanizm, który posiada silnik, przekładnię oraz sterownik w jednej obudowie. Zadaniem tego mechanizmu jest poruszanie zamontowanym ramieniem wraz jednoczesnym zapamiętaniem swojego położenia. Kod odpowiedzialny za sterowanie drzwiami został zamieszczony w tabeli 3.

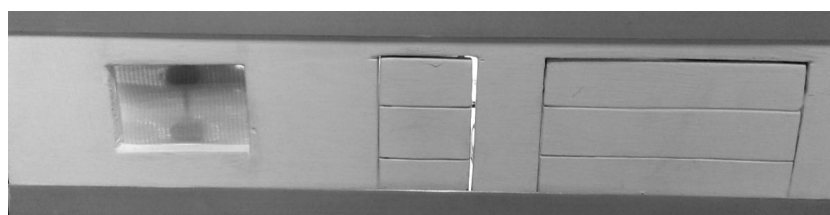
Tabela 3. Fragment kodu Arduino odpowiedzialny za sterowanie drzwiami

<pre>#include&lt;Servo.h&gt; Servo drzwi; drzwi.attach(2); if (string == "OD") {   drzwi.write(180);</pre>	<pre>delay(15); } if (string == "ZD") {   drzwi.write(45);   delay(15); }</pre>
--	---

Na rys. 6 i rys. 7 został przedstawiony stan działania serwomechanizmów, które w projekcie są odpowiedzialne za otwieranie drzwi, rolet oraz bramy garażowej.

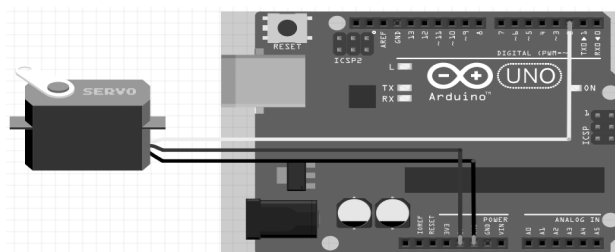


Rys. 6. Serwomechanizmy otwarte



Rys. 7. Serwomechanizmy zamknięte

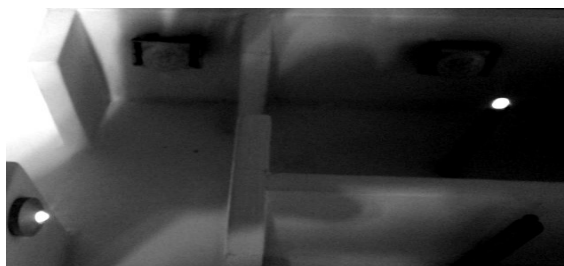
Serwomechanizmy zostały tak zaprogramowane, aby połączone do nich ramiona odchyłały się o odpowiedni kąt. Całkowity możliwy kąt odchylenia w przypadku zastosowanych serwomechanizmów wynosi 180°. Schemat podłączenia serwomechanizmu do środowiska Arduino UNO został przedstawiony na rys. 8.



Rys. 8. Schemat podłączenia serwomechanizmu do Arduino UNO v3

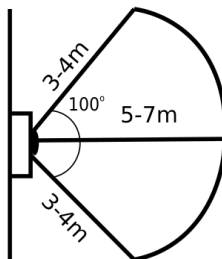
### 4.3. Czujnik ruchu

Czujnik ruchu po wykryciu ruchu wysyła sygnał do platformy programistycznej Arduino, która następnie uruchamia odpowiednią diodę LED. Cały proces jest wysoce konfigurowalny w takich aspektach jak np. odpowiednie opóźnienie, zakres sensorów czy też tryb wykrywania ruchu. Weryfikacja działania czujników ruchu została przedstawiona na rys. 9.



Rys. 9. Wykrycie ruchu poprzez czujniki, które skutkuje zaświeceniem się diod LED

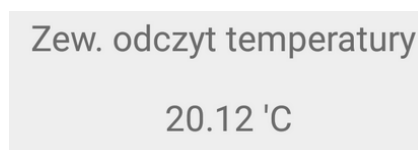
Czujniki ruchu zastosowane w pracy posiadają regulację zasięgu wykrycia obiektu w granicach od 3 m do 7 m i kąta widzenia do  $100^\circ$  (rys. 10). Dodatkowo oprócz regulacji zakresu widzenia można zmienić czas trwania stanu wysokiego po wykryciu obiektu (od 5s do 200s).



Rys. 10. Zakres widzenia zastosowanego czujnika ruchu

#### 4.4. Czujnik temperatury

Na rys. 11 został przedstawiony stan odczytu z cyfrowego czujnika temperatury (DS18B20) podłączonego do Arduino. Odczytana temperatura wyświetla się na ekranie urządzenia mobilnego z zainstalowaną aplikacją.



Rys. 91. Odczyt wartości z czujnika temperatury

W tabeli 4 został zaprezentowany fragment kodu odpowiedzialny za obsługę tego czujnika.

Tabela 4. Fragment kodu oprogramowania Arduino odpowiedzialnego za odczyt bieżącej temperatury otoczenia

```
#include <OneWire.h>
#include <DS18B20.h>

// Numer pinu do którego podłączasz czujnik
#define ONEWIRE_PIN 5

// Adres czujnika
byte address[8] = {0x28, 0xFF, 0x87, 0x7E, 0x0, 0x16, 0x2, 0x95};
OneWire onewire(ONEWIRE_PIN);
DS18B20 sensors(&onewire);

if (sensors.available())
{
    float temperature = sensors.readTemperature(address);
    Serial.print(temperature);
    Serial.println(F(" 'C"));
    sensors.request(address);
}
```

## 5. PODSUMOWANIE

Założeniem przedstawionej pracy było utworzenie możliwie taniego systemu sterującego inteligentnym domem przy użyciu technologii mobilnych oraz platformy Arduino.



Autorzy założyli także konieczność spełnienia takich wymogów przez projekt jak: łatwość uwzględnienia innych obiektów sterowania w inteligentnym domu, łatwość konfigurowalności procesu sterowania w takich aspektach jak np. odpowiednie opóźnienie, zakres sensorów czy też tryb wykrywania ruchu, kąta otwarcia drzwi, itp..

Podstawową zaletą przedstawionego w pracy projektu jest więc stosunek kosztów, które są niewielkie w porównaniu do szerokiego zakresu możliwości środowiska Arduino.

W projekcie została przedstawiona tylko część rozwiązań, które proponuje Arduino UNO v3, co czyni projekt otwartym na nowe udoskonalenia.

Aktualnie, realizowana będzie kolejna faza rozwoju projektu, tzn. sterowanie za pomocą komend głosowych (zastosowanie modułu rozpoznawania głosu) lub rozpoznawanie gestów i zastosowanie do nich odpowiedniej funkcjonalności.

#### LITERATURA

- [1] Monk S.: *Arduino dla początkujących. Podstawy i szkice*, HELION, Gliwice, 2014.
- [2] Clements–Croome D.: *Intelligent Buildings: design, management and operation*”, Thomas Telford LTD, 2004.
- [3] Conder S., Darcey L.: *Android Programowanie aplikacji na urządzenia przenośne – wydanie II*, HELION, Gliwice, 2011.
- [4] Wiązania M.: *Kurs Arduino Język Programowania*, „Elektronik”, 4, 2011, s. 98–102.
- [5] <http://www.arduino.cc/>
- [6] *The LonWorks Platform: Technology Overview*”, <http://www.echelon.com/developers/lonworks/default.htm>

#### CONTROLLING INTELLIGENT HOME USING A MOBILE APPLICATION THAT USES A MICROCONTROLLER ARDUINO UNO V3

This article describes control system for smart home of the mobile application for Android system and microcontroller Arduino UNO v3. It was created a model simulating the operation of the intelligent home and developed software for mobile devices with Android system and software for microcontroller Arduino UNO in the C programming language. To control smart home has been used Bluetooth Technology.

(Received: 07. 02. 2017, revised: 28. 02. 2017)