

# **SPRAWOZDANIE KOŃCOWE PT: BUDOWA SYSTEMU KUCHENNEGO DLA OSÓB NIEPEŁNOSPRAWNYCH.**

## **Opracowali:**

Robert Sasal

Grzegorz Dygas

Tomasz Napora

## **Spis treści**

Adaptacja systemu drzwi uchylnych :.....	S. X
Adaptacja systemu odsuwania szuflad .....	S. X
Adaptacja systemu opuszczania szafek :	..... S. X
Adaptacja systemu kogeneracji :	..... S. X
Adaptacja sterowania całością systemu :	..... S. X
Adaptacja aplikacji sterującej .....	S. X
Instrukcja obsługi aplikacji sterującej .....	S. X

## **Adaptacja systemu drzwi uchylnych**

W ramach prac dokonano adaptacji zakupionego systemu otwierania drzwi uchylnych. Adaptacji wymagał system bezpieczeństwa, oraz mocowanie siłownika i jego sterowanie.

Modyfikacji wystawiania dokonano za pomocą zmiany kodu aplikacji sterującej.

Natomiast modyfikacji technicznej dokonano podczas procesu montażu, co przedstawiają poniższe zdjęcie.



zdj 1. Mechanizm otwierania drzwi

W ramach dodatkowej funkcyjności, w systemie otwierania drzwi wprowadzono funkcję blokady rodzicielskiej. Blokada ta pozwala z poziomu aplikacji możliwości otwarcia szafki od zewnątrz. Uniemożliwia tym samym dostęp do wnętrza szafki dzieciom lub osobom niepożądanym



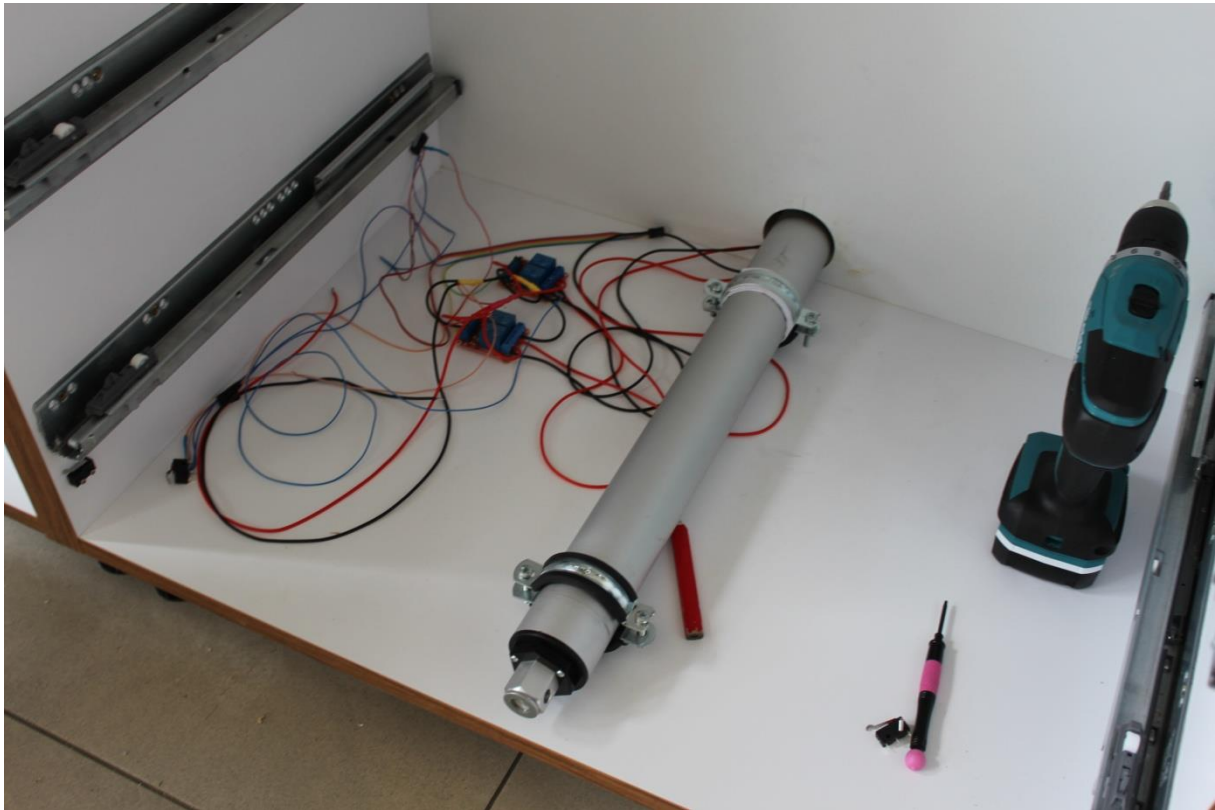
zdj 2. Widok ogólny mechanizmu otwierania i zamykania szafki.

## **Adaptacja systemu odsuwania szuflad**

W ramach prac dokonano adaptacji zakupionego systemu odsuwania szuflad. W przeciwieństwie do rozwiązania zakupionego, zmodyfikowano umieszczenie siłownika. Ze względów praktyczno - estetycznych, zdecydowano o zamontowaniu napędu pod szufladą i wykonanie szuflady o płytszym dnie. Takie rozwiązanie sprawia, że system odsuwania szuflady jest kompletnie niewidoczny. Poniższe zdjęcia przedstawiają system podczas montażu.



zdj 3. Zamontowany siłownik, wraz z widocznymi prowadnicami szuflady



zdj 4. Sterowanie siłownika podczas montażu

Przy adaptacji systemu niezbędne było określenie długości na jaką wysuwa się szuflada. W tym celu zastosowano styczniki które po załączeniu przekazują do jednostki sterującej informację o krańcowych pozycjach szuflady. Takie rozwiązanie zapewnia dużą uniwersalność urządzenia i jedynym ograniczeniem jest długość na jaką wysuwa się siłownik. Planuje się wykonać odlewane plastikowe wskaźniki do styczników krańcowych. Na etapie montażu prototypowego rozwiązania zabudowy kuchennej, wykonano wskaźnik drewniany, widoczny na poniższym zdjęciu.

## Adaptacja systemu opuszczania szafek

W ramach prac dokonano adaptacji zakupionego od Politechniki Warszawskiej systemu opuszczania. System został zabudowany w specjalnie do tego celu zbudowanej szafce meblarskiej. Stelaż szafki wykonano z tradycyjnych płyt wiórowych, natomiast drzwi i półki szafki wykonano z opracowanych w ramach niniejszych prac płyt meblarskich o ograniczonej masie. Poniższe zdjęcie przedstawia zabudowę stelaża w korpusie szafki meblarskiej.



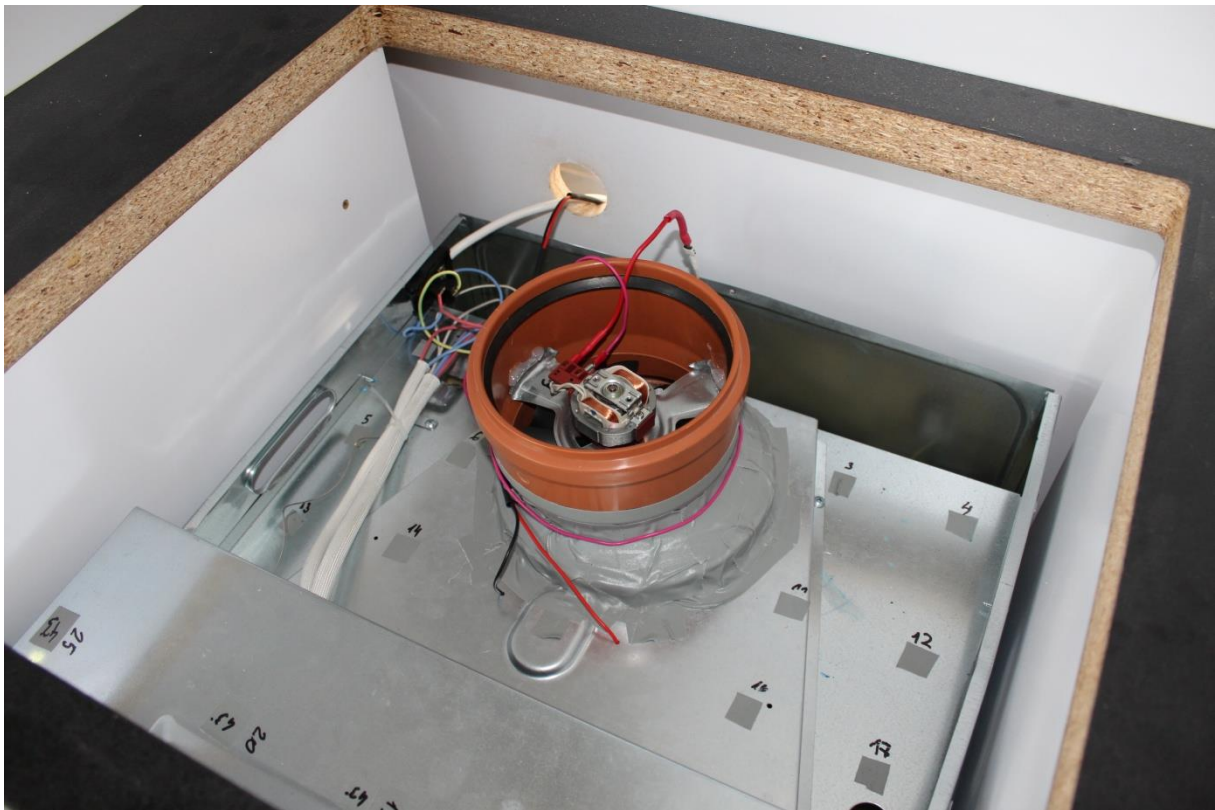
zdj 5. Zabudowa stelaża podnoszenia/opuszczania w korpusie szafki meblarskiej.

Założeniem autorów konstrukcji stelaża była jego uniwersalność, umożliwiającą dostosowanie go w szerz (wymiana poprzeczek) jak i wzwyż (regulacja wysokości wysuwanych belek). Jest to rozwiązanie dobre w przypadku próby montażu stelaża do gotowej szafki. W naszym przypadku możliwe jest wykonanie szafki pod konkretne wymagania wymiarowe. W związku z powyższym dokonano modyfikacji mocowania, usuwając regulowane poprzeczki. Ich usunięcie umożliwiło spłylenie zabudowy stelaża, a co za tym idzie, większa powierzchnia wewnątrz szafki przeznaczona jest do wykorzystania. Zmodyfikowany aluminiowy stelaż szafki przedstawia poniższe zdjęcie.

zdj 6. Płyta umieszczona w dolnej części szafki opuszczanej.

## Adaptacja systemu kogeneracji

W ramach prac dokonano adaptacji zakupionego systemu kogeneracji. W demonstracyjnej zabudowie kuchennej zabudowano piekarnik wyposażony w ogniwa odzyskujące energię elektryczną z układu chłodzenia obudowy. Obecnie praktycznie każdy piekarnik wyposażony jest w system nadmuchu na rączkę zimnym powietrzem, które jednocześnie chłodzi górną część obudowy. Zdjęcie odkrytej obudowy piekarnika przedstawiono poniżej.



zdj 7. Górna część piekarnika z systemem odzyskiwania energii elektrycznej.

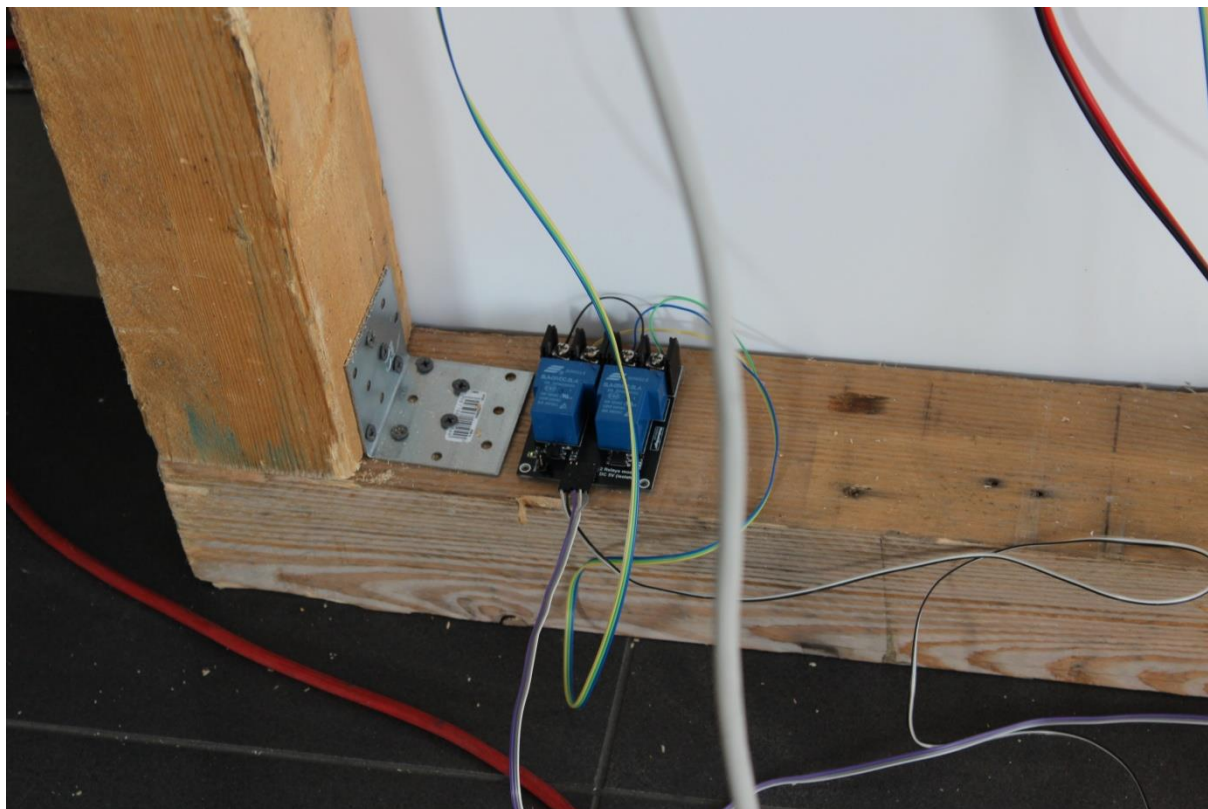
Dodatkowo w korpusie szafki wykonano specjalne kanały powietrzne doprowadzające ładunek świeżego powietrza z dolnej części zabudowy (kratka w dolnej osłonie przypodłogowej).

Kolejnym elementem systemu kogeneracyjnego było umieszczenie paneli fotowoltaicznych na dachu budynku.



## Adaptacja sterowania całością systemu

W ramach prac dokonano adaptacji układów sterujących całością systemu. Dokonano wymiany na większe przekaźników prądowych, tak aby zapewnić ich znaczące przewymiarowanie, a co za tym idzie maksymalnie wysoką bezawaryjność całości systemu. Przekaźniki podczas montażu przedstawiono na poniższym zdjęciu.



zdj 8.      Przekaźniki podczas montażu



zdz 9. Całość sterowania systemem



zdj 10. Szafka w pozycji otwartej.

## **Adaptacja aplikacji sterującej**

### *1. Zakres zmian w aplikacji wprowadzonych celem dostosowania do współpracy ze sterownikiem Arduino.*

Aplikacja wymagała dostosowania do współpracy z wykorzystanym w projekcie sterownikiem Arduino.

Wprowadzono zmiany w zakresie:

- protokołu komunikacji,
- formatu komunikatów,
- częstotliwości wysyłania zapytań o status.

Wprowadzono również kilka poprawek, m. in.

- obsługę wyjątków w przypadku braku dostępnych sieci WiFi,
- odświeżanie ekranu - stan szafek
- obsługę oświetlenia.

W zakresie panelu użytkownika nie wprowadzono zmian.

W dalszej części opisano jedynie zmiany wprowadzone a kodzie źródłowym aplikacji podczas prac związanych z dostosowaniem jej do współpracy ze sterownikiem Arduino.

Nieopisane w tym zestawieniu elementy programu nie uległy zmianie.

## 2. Struktury danych

Aplikacja wykorzystuje zorganizowane struktury danych do przechowywania i przetwarzania danych związanych z układem szafek i oświetlenia oraz do wymiany informacji ze sterownikiem zewnętrznym. Struktury te to:

### Plik konfiguracyjny

**Plik konfiguracyjny jest tworzony przez zespół montujący system w kuchni i nie jest edytowalny przez użytkownika.**

Składa się z dwóch modułów - nagłówka pliku oraz  $n$  elementów, po jednym dla każdego elementu systemu. Struktura nagłówka nie została zmieniona.

Dla każdego z elementów kuchni tworzy się jeden obiekt klasy `PskConfigFileItem`, zawierający szczegółowe dane o każdym z elementów systemu. W ramach dostosowania dodano pole: `itemPrevState`, które przechowuje dane o poprzednim stanie szafki. Jest ono wykorzystywane w przypadku wznowiania pracy po anulowaniu operacji otwierania lub zamykania, gdy element znajduje się w stanie idle.

```
public class PskConfigFileItem
{
    public int id;
    public string relayForward;
    public string relayBackward;
    public string edgeForward;
    public string edgeBackward;
    public int safetyPin;
    public string pinoutLight;
    public bool parentalLock;
    public string type { get; set; }
    public int xCoord;
    public int yCoord;
    public int width;
    public int height;
    public string itemState { get; set; }
    public string itemPrevState { get; set; } = "";
    public bool ledState { get; set; }
}
```

Dodatkowo zmieniono typ `itemState` na `string`. W bieżącej konfiguracji stan elementu jest przechowywany tekstowo, zgodnie z dokumentacją dostarczoną od projektanta oprogramowania zaimplementowanego na sterowniku Arduino.

## Protokół transmisji danych

Ze względu na możliwości oraz złożoność programowania sterownika Arduino, zrezygnowano z przesyłania komunikatów w formacie JSON, na rzecz ciągu tekstowego par klucz - wartość.

Wprowadzono w kodzie aplikacji mechanizm konwertujący otrzymywane od sterownika dane na format JSON:

```
fileContent = "[" + fileContent + "];  
  
fileContent = fileContent.Replace("%&", "\", \"val\": \"");  
fileContent = fileContent.Replace("&", "\", {\"key\": \"");  
fileContent = fileContent.Replace("[%", "[{ \"key\": \"");  
fileContent = fileContent.Replace("&]", "\", }]");
```

W wyniku działania powyższej funkcji otrzymano dane w formacie JSON.

### 3. Komunikacja sieciowa

Aplikacja wykorzystuje do komunikacji ze sterownikiem systemu protokół UDP.

Komunikaty są przesyłane z wykorzystaniem zdefiniowanych portów (zmienionych w wyniku dostosowania aplikacji na: 61550 i 61551).

#### **Komunikacja inicjująca pracę systemu - po dostosowaniu do współpracy ze sterownikiem Arduino**

Po uruchomieniu aplikacja wysyła komunikat hello sformatowany w JSON, którego celem jest wywołanie funkcji wysyłającej plik konfiguracyjny ze sterownika.

hello\_0

W odpowiedzi aplikacja otrzymuje informację o liczbie części pliku konfiguracyjnego:

2

Następnie aplikacja wywołuje funkcję hello, celem otrzymania kolejnych fragmentów pliku konfiguracyjnego:

hello\_1

hello\_2

otrzymując w odpowiedzi *i*-ty fragment:

```
%liczbaSzaf%&4%&3_isLED%&TRUE%&3_type%&LED%&3_relayForward%&47%&0_relayForward%&24%&0_relayBackward%&22%&0_edgeForward%&23%&0_edgeBackward%&25%&0_safetyPin%&27%&0_type%&szuflada%&0_xCoord%&100%&0_yCoord%&100%&1_relayForward%&28%&1_relayBackward%&30%&1_edgeForward%&29%&1_edgeBackward%&31%&1_safetyPin%&33%&2_relayForward%&34%&2_relayBackward%&36%&2_edgeForward%&35%&2_edgeBackward%&37%&2_safetyPin%&39&
```

kontynuując komunikację aż do uzyskania `i = numOfItems` elementów pliku.

Następnie zawartość pliku jest łączona w jedną całość, formatowana do JSON i zamieniana na obiekty `PskConfigFileItem`.

#### **Przesyłanie komunikatów wywołujących rozkazy akcji - po dostosowaniu do współpracy ze sterownikiem Arduino**

Komunikaty wywołujące obsługę zdarzeń / akcji wysyłane są z wykorzystaniem `id` - oznaczającym identyfikator szafki lub lampki która ma zostać uaktywniona oraz słowa kluczowego, np.:

2\_forward

## **Cykliczne testowanie połączenia - po dostosowaniu do współpracy ze sterownikiem Arduino**

Aplikacja cyklicznie sprawdza, czy sterownik zewnętrzny jest obecny w sieci, wysyłając komunikat appPresent . Jeśli przez 5 sekund nie otrzyma odpowiedzi, interpretuje ją jako zerwanie połączenia.

Komunikat App wygląda następująco:

status

Aplikacja powinna otrzymać w odpowiedzi ze sterownika w kolejnych pakietach UDP:

0\_idle

1\_backward

2\_opened

3\_idle