

PRACE
KÓŁ
NAUKOWYCH
Politechniki
Rzeszowskiej
w roku
akademickim
2021/2022

Tom 1

Rzeszów 2022

Wydano za zgodą Rektora

R e d a k t o r n a c z e l n y
Wydawnictw Politechniki Rzeszowskiej
dr hab. inż. Lesław GNIEWEK, prof. PRZ

Przewodniczący Rady Redakcyjnej
Bartosz TRYBUS

Rada Redakcyjna
Prac Kół Naukowych 2021/2022
Dorota GŁOWACZ-CZERWONKA

Paweł KUT
Mariusz NYCZ
Bartosz PAWŁOWICZ
Mirosław SOBASZEK
Bartosz TRYBUS

W procesie wydawniczym pominięto etap opracowania językowego.

Skład i łamanie
Joanna MIKUŁA

Bezzałogowy statek powietrzny, PM10, PM25, PM5, BSP, dron do badania jakości powietrza, Angular.ts, Python, RaspberryPi4b, aplikacja desktopowa, Qt5 framework, ESP32, RFID, NFC, C++, Wyświetlacz LED, Matryca LED, PIC32MX, SPI, MAX7219, onion routing, The Tor Project, Tor Browser, sztywne pianki poliuretanowe, właściwości fizyczne, gęstość pozorna, stabilność wymiarów, chłonność wody, współczynnik przewodzenia ciepła, wentylacja, wilgotność, system operacyjny, sterowniki parawirtualizowane, interfejs programistyczny, virti i in.

© Copyright by Oficyna Wydawnicza Politechniki Rzeszowskiej
Rzeszów 2022

e-ISBN 978-83-7934-607-3
p-ISBN 978-83-7934-597-7

Oficyna Wydawnicza Politechniki Rzeszowskiej
al. Powstańców Warszawy 12, 35-959 Rzeszów
<https://oficyna.prz.edu.pl>

Ark. wyd. 8,93. Ark. druk. 12,00. Wydrukowano we wrześniu 2022 r.
Drukarnia Oficyny Wydawniczej PRZ, al. Powstańców Warszawy 12, 35-959 Rzeszów
Zam. nr 47/22

SPIS TREŚCI

STUDENCKIE KOŁO NAUKOWE ELEKTRONIKI I TECHNOLOGII INFORMACYJNYCH

Rafał NAZARKO, Piotr STOREK, Karol SIWIEC, Marek GIŻA, Bartosz PAWŁOWICZ Bezzałogowy statek powietrzny do badania zanieczyszczenia powietrza	7
Rafał NAZARKO, Piotr STOREK, Karol SIWIEC, Marek GIŻA, Bartosz PAWŁOWICZ System identyfikacji zawodników oraz dokonywania pomiarów czasu trwania przejazdów na zawody ROBO-MOTION	17
Rafał NAZARKO, Piotr STOREK, Karol SIWIEC, Marek GIŻA, Bartosz PAWŁOWICZ Re-pet ender czyli maszyna do przetwarzania butelek pet na filament	29
Rafał NAZARKO, Piotr STOREK, Karol SIWIEC, Marek GIŻA, Bartosz PAWŁOWICZ Uniwersalny komputer pokładowy	39

STUDENCKIE KOŁO NAUKOWE ELEKTRONIKÓW

Patryk JASKUŁA, Hubert SKALSKI, Łukasz SZKARADEK, Krzysztof SADOWY Mirośław SOBASZEK Wyświetlacz przemysłowy oparty na matrycach LED	53
Łukasz SZKARADEK, Krzysztof SADOWY, Patryk JASKUŁA, Hubert SKALSKI, Mirośław SOBASZEK Obsługa tensometru za pomocą mikrokontrolera STM32 na przykładzie wagi stołowej	67

STUDENCKIE KOŁO NAUKOWE SOMETHING ABOUT IT SECURITY

Marek PIĘDEL, Mariusz NYCZ Jak cebula wpływa na prywatność użytkownika w Internecie	81
--	----

STUDENCKIE KOŁO NAUKOWE „ESPRIT”

Klaudia STACHNIK, Dorota GŁOWACZ-CZERWONKA Uniepalnione, sztywne pianki poliuretanowe na bazie Rokopolu RF 151V. Cz. 1. Właściwości fizyczne	93
Ewelina SZOSTEK, Dorota GŁOWACZ-CZERWONKA Badanie palności pianek poliuretanowych na bazie Rokopolu RF-551 z dodatkiem wybranych uniepalniaczy addytywnych	107
Ewelina SZOSTEK, dr inż. Dorota GŁOWACZ-CZERWONKA Antypireny i ich rola w uniepalnianiu pianek poliuretanowych	115
Klaudia STACHNIK, Dorota GŁOWACZ-CZERWONKA Uniepalnione, sztywne pianki poliuretanowe na bazie Rokopolu RF 151V. Cz. 2. Badanie palności kompozycji	123

STUDENCKIE KOŁO NAUKOWE HVAC Ogrzewanie, Wentylacja, Klimatyzacja

Justyna KUNA, Agata OLEJARCZUK, Oskar RUMAK, Paweł KUT Pomiary skuteczności wentylacji mechanicznej wywiewnej w budynkach użyteczności publicznej	137
--	-----

STUDENCKIE KOŁO NAUKOWE INFORMATYKÓW „KOD”

Szymon RAPAŁA, Bartosz TRYBUS Sposoby autoryzacji dostępu do danych w aplikacjach wykorzystujących REST API	153
Jakub PRZYSTASZ, Bartosz TRYBUS Sterowniki parawirtualizowane w systemie MicrOS	163
Oskar TYNIEC, Bartosz TRYBUS Zastosowanie biblioteki React na przykładzie aplikacji do zarządzania za- daniami	177



KOŁO

NAUKOWE

○ ELEKTRONIKI

I TECHNOLOGII

INFORMACYJNYCH



Rafał NAZARKO, Piotr STOREK, Karol SIWIEC, Marek GIŻA

dr inż. Bartosz PAWŁOWICZ
opiekun naukowy

BEZZAŁOGOWY STATEK POWIETRZNY DO BADANIA ZANIECZYSZCZENIA POWIETRZA

Artykuł opisuje sposób wykonania i zasadę działania systemu bezzałogowego statku powietrznego do badania jakości powietrza. W ramach artykułu został wykorzystany bezzałogowy statek powietrzny typu hexacopter (statek powietrzny, który wyposażony jest w sześć ramion oraz sześć specjalnie dobranych silników). Jednostka wykorzystuje niezawodny system wraz z czujnikami niezbędnymi do stabilnego lotu w trybie GPS (Global Positioning System) to wspomaganie sterowania z wykorzystaniem dodatkowych odbiorników gps. Bezzałogowy statek powietrzny dzięki wykorzystaniu tego typu czujników może pozostać stabilny i nieruchomy nawet przy niekorzystnych warunkach pogodowych. Stabilność oraz unieruchomienie drona jest bardzo ważnym aspektem w kontekście badania jakości powietrza. Dron jest w stanie zawisnąć nad wybranym przez pilota punkcie i dokładnie prześledzić wyniki pomiarów na aplikacji internetowej napisanej w technologii react.js. Poszczególne punkty opisują budowę oraz zastosowane technologie przy konstruowaniu statku powietrznego do badania jakości powietrza. Statek powietrzny podzielony jest na dwa niezależne od siebie moduły. Moduł kontroli lotu dzięki któremu widzimy telemetrię w czasie rzeczywistym oraz niezależny moduł kontroli jakości powietrza umieszczony pod spodem bezzałogowa zaprojektowany i wydrukowany w technologii druku 3D. Główną jednostką, która przetwarza dane i wysyła je do skonfigurowanej bazy danych jest mikrokomputer Raspberry Pi 4b. Mikrokomputer jest przewodowo połączony z routerem WIFI, który zawiera w sobie moduł 4G LTE dzięki któremu możliwa jest dwustronna komunikacja z jednostką pomiarową. Dodatkowo w projekcie zastosowano możliwość komunikacji radiowej wraz z RFD 868 TXMOD V2, który jest systemem telemetrii i komunikacji na daleki zasięg, składający się z modemu radiowego nadajnika TXMOD V2. Dzięki zastosowaniu dodatkowej komunikacji radiowej pilot bezzałogowego statku powietrznego ma możliwość wykonywania operacji dronowej w przestrzeni otwartej nawet do odległości 40km według dokumentacji producenta.

Słowa kluczowe: Bezzałogowy statek powietrzny, PM10, PM25, PM5, BSP, Dron do badania jakości powietrza, Angular.ts, Python, RaspberryPi4b.

WPROWADZENIE

Na przełomie ostatnich lat coraz większy udział na rynku stanowią bezzałogowe statki powietrzne. Popularność dronów wzrasta przez wzgląd na coraz łatwiejsze możliwości sterowania oraz na bardzo dużą liczbę zastosowań. Swoje zadania wykonują pomagając ratownikom medycznym przy poszukiwaniu zaginionych osób szczególnie w terenach wysokogórskich, dzięki użyciu specjalnych kamer termowizyjnych o dużej rozdzielczości, w rolnictwie jako źródła do zapobiegania przesuszeniu dzięki użyciu kamer multispektralnych, w wojsku do lokalizowania i likwidowania celów czy w przemyśle do monitorowania stanu linii energetycznych.

Celem projektu jest automatyczny dron dalekiego zasięgu, badający stężenie zanieczyszczonego powietrza, wykorzystujący czujniki pyłów PM_{2,5} i PM₁₀. Projektowane urządzenie ma na celu ułatwienie wykonywania pomiarów nad daną przestrzenią powietrzną przy jednoczesnym zwróceniu uwagi na problem smogu na Ziemi. Ułatwienie obsługi drona polega na zastosowaniu automatycznego sterowania. Zawarty w budowie drona GPS (Global Positioning System) wraz z telemetrią i ardupilotem pozwala na precyzyjne wyznaczenie trasy do badania przelotu i zbadania jakości powietrza.

Inspiracją do stworzenia bezzałogowego statku powietrznego badającego stężenie pyłów PM₁₀ i PM_{2,5} jest coraz gorszy stan jakości powietrza w naszym otoczeniu. Charakter projektu ma postać naukową, nie przeznaczoną na komercjalizację. Wykonany projekt ma służyć do pomiarów jakości powietrza na dużym obszarze bez potrzeby umieszczania wielu pojedynczych czujników. W większości przypadków ludzie nie zdają sobie sprawy z powagi sytuacji, co stanowi ogromne zagrożenie dla ich zdrowia a nawet życia. Podejmowane tego typu działania przyczyniają się do edukacji społeczeństwa, pogłębiają jego świadomość na temat zanieczyszczenia powietrza.

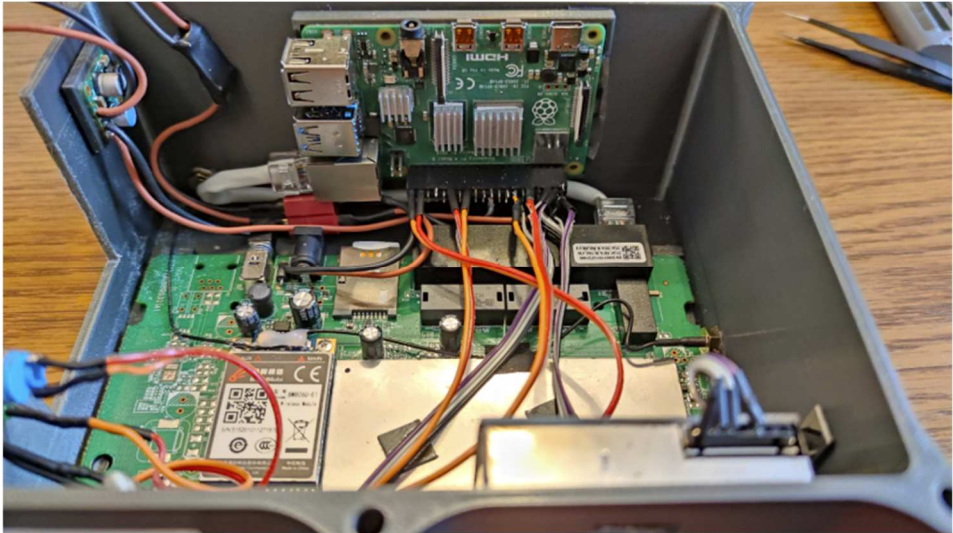
CZUJNIK PM 2,5 mierzy aerozole atmosferyczne o średnicy nie większej niż 2,5µm (mikrometra). Zdaniem Światowej Organizacji Zdrowia są one najbardziej szkodliwe dla zdrowia człowieka spośród innych zanieczyszczeń atmosferycznych. Jak wynika z raportów Światowej Organizacji Zdrowia (WHO), długotrwałe narażenie na działanie pyłu zawieszonego PM_{2,5} skutkuje skróceniem średniej długości życia. Krótkotrwała ekspozycja na wysokie stężenia pyłu PM_{2,5} powoduje wzrost liczby zgonów spowodowanych chorobami układu oddechowego oraz wzrost ryzyka nagłych przypadków wymagających hospitalizacji (nasilenie astmy, ostra reakcja układu oddechowego, osłabienie czynności płuc), gdyż tak drobny pył dostaje się bezpośrednio przez płuca do krwi.

CZUJNIK PM₁₀ wskazuje mieszaniny zawieszone w powietrzu, są to cząsteczki o średnicy 10µm. W ich skład mogą wchodzić takie substancje toksyczne jak np. benzopireny, dioksyny i furany. Występowanie pyłów PM₁₀ związane jest między innymi z procesami spalania paliw stałych i ciekłych.

1. BUDOWA BEZZAŁOGOWEGO STATKU POWIETRZNEGO (HEXACOPTER)

Bezzałogowy statek powietrzny został zbudowany z ogólnodostępnych elementów elektronicznych i konstruktorskich. Rama, która została wybrana w ramach realizacji projektu to Tarot FY680 PRO, która została zaprojektowana do budowy dronów które docelowo posiadają sześć niezależnych silników. Rama wielowirnikowca wykonana jest z włókna węglowego dzięki temu waży niecałe 580g. Wielowirnikowiec wyposażony jest także w sześć silników EMAX MT3510-600KV wraz z specjalnie dobranymi regulatorami DJI E600 30A idealnie do parametrów silnika. Kontroler lotu, który jest na pokładzie bezzałogowca to Pixhawk 2.1 Cube Orange wraz z modułem GPS Here GNSS 3.0 CAN. Pilot ma możliwość wyboru sterowania bezzałogowcem. W przypadku gdy chcę latać na bliskie odległości i mieć przy tym zwiększoną precyzję może skorzystać z zamontowanego odbiornika FRSKY L9R, jednakże gdy planuje loty na dłuższe dystanse może przełączyć się na system telemetrii dalekiego zasięgu RFD 868x komunikujący się z bezzałogowcem po protokole PPM (PulsePositionModulation). Elementy zasilające komponenty to akumulator Multistar 4S 14,8V o pojemności 7000mAh co pozwala na lot o długości od piętnastu do trzydziestu minut w zależności od warunków atmosferycznych.

Zewnętrznie specjalnie zaprojektowana i wydrukowana w technologii druku 3D skrzynia pomiarowa zawiera w sobie elementy zarówno pomiarowe jak i elementy niezbędne do zbierania bądź archiwizacji danych. Do pomiaru wilgotności, temperatury oraz ciśnienia został wybrany czujnik BME680 podłączony do głównego mikrokomputera Raspberry Pi 4b zasilany napięciem 3.3V. Zakres pomiarowy: temperatura: -40 °C do 85 °C, wilgotność: 10 %RH do 100 %RH, ciśnienie: 300 hPa do 1100 hPa. Do pomiaru pyłków powietrza został użyty laserowy czujnik pyłu umożliwiający monitorowanie czystości powietrza. Pozwala on wykryć cząstki o średnicy powyżej 0,3 μm, przeznaczony do pomiaru wartości PM2.5 i PM10. Posiada wentylator zasysający powietrze. Układ zasilany jest napięciem od 4,7 V do 5,3 V, komunikuje się poprzez sygnał UART lub PWM. Jako mikrokomputer do zarządzania całym procesem zbierania danych o zanieczyszczeniu powietrza został użyty popularny mikrokomputer Raspberry pi dzięki, któremu możemy bezpośrednio zrealizować połączenie z bazą danych poprzez modem 4G LTE, który opcjonalnym elementem w routerze. Mikrokomputer jest połączony z routerem poprzez kabel sieciowy LAN UTP KAT 5E. Dzięki zastosowaniu połączenia trwałego cały układ jest bardziej odporny na zakłócenia oraz wpływy zewnętrzne.



Rysunek 1. Moduł pomiarowy skrzyni zawierającej komponenty pomiarowe
Źródło: archiwum własne

2. KOD PROGRAMU WGRANY NA MIKROKOMPUTER RASPBERRY PI

```
class database_handler:
def init (self):
self.db = pymysql.connect(host='X.X.X.X', user='root',
password='XXXXXXX', database='dron_smogowy', charset='utf8mb4', cur-
sorclass=pymysql.cursors.DictCursor, ssl_ca= '/custom/location/.ca,
ssl_cert = '/custom/location/.pem, ssl_key = '/custom/location/.pem')

def insert_values_measurement(self,db_name,IdMeasurement,pm10,pm25,hu-
midity,hPa,temperature,IAQ,time): mycursor = self.db.cursor()
query = 'INSERT INTO dron_smogowy.{}(IdMeasurement,pm10,pm25,humid-
ity,hPa,
temperature,IAQ,time) VALUES (%s,%s, %s, %s, %s, %s, %s, %s)'.for-
mat(db_name)
val = (IdMeasurement,pm10,pm25,humidity,hPa,temperature,IAQ,time) my-
cursor.execute(query,val)
self.db.commit()
print(mycursor.rowcount, "record(s) inserted.")

def insert_values_localization(self,tb_name,id,longtitude,lati-
tude,satelite):
```

```

mycursor = self.db.cursor()
query = 'INSERT INTO dron_smogowy.{}(IdMeasurement, longitude, latitude) VALUES(%s, %s, %s)'.format(tb_name)
val = (id,longitude,latitude)
mycursor.execute(query,val)
self.db.commit()
print(mycursor.rowcount, "record(s) inserted into localization.")

def read_specific_value(self,chosen_value):
mycursor = self.db.cursor()
query='SELECT {} dron_smogowy.measurement'.format(chosen_value) mycursor.execute(query)
rows = mycursor.fetchall()

def create_new_table_into_db(self,name):
mycursor = self.db.cursor() print(name)
query='CREATE TABLE {} (IdMeasurement INT AUTO_INCREMENT PRIMARY KEY, pm10 FLOAT, pm25 FLOAT, humidity FLOAT, hPa FLOAT, temperature FLOAT, IAQ FLOAT, time TIMESTAMP)'.format(name) print(query)
mycursor.execute(query)

def save_values_measurement_to_json_file(self,database_name): mycursor = self.db.cursor()
query = 'SELECT * FROM dron_smogowy.{}'.format(database_name) mycursor.execute(query)
rows = mycursor.fetchall()

object_list = [] for x in rows:
print(x)
d = collections.OrderedDict()
d["pm10"] = x.get("pm10")
d["pm25"] = x.get("pm25")
d["humidity"] = x.get("humidity")
d["hPa"] = x.get("hPa") d["temperature"] = x.get("temperature")
d["IAQ"] = x.get("IAQ")
d["time"] = x.get("time").strftime('%Y-%m-%d %H:%M:%S') object_list.append(d)

result = {"measurements":object_list} j = json.dumps(result,indent=7)
with open(self.path_json_file, "w") as f: f.write(j)

```

Listing 1. Kod programu wgrany na mikrokontroler Raspberry

Przedstawiony fragment kodu opisuje klasę, która jest elementem wykorzystywanym w głównym bloku programu. Klasa jest inicjowana raz przy momencie uruchamiania głównego bloku programu. Klasa inicjuje połączenie poprzez połączenie z wybranym hostem, na którym hostowany jest serwis bazodanowy. W tym przypadku postawiliśmy na otwarty-źródłowy serwis bazodanowy MySQL

w wersji 5.5. Za komunikację i migrację danych odpowiada skrypt Python'owy, który dzięki kluczom przy użyciu certyfikatu SSL migruje dane z bazy GCP. (Google Cloud Platform). Szyfrowanie przy użyciu certyfikatu SSL opiera się na dwóch kluczach: prywatnym i publicznym. Klucz jest ciągiem znaków alfanumerycznych, wygenerowanych za pomocą odpowiedniego algorytmu. Za jego pomocą odbywa się szyfrowanie i deszyfrowanie danych w komunikacji SSL. Baza danych oparta jest o usługę Google Cloud Platform. To usługa bazy danych w chmurze, czyli database-as-a-service (DBaaS). Dane w bazie danych są przechowywane i przetwarzane w chmurze, na infrastrukturze usługodawcy chmurowego, a dostęp jest zapewniany z poziomu konsoli Google Cloud Platform lub wiersza poleceń.

Metoda `insert_values()` odpowiada za wstawianie poszczególnych rekordów do bazy danych. Dzięki zastosowaniu formatowania przy każdym nowym połączeniu z bazą danych tworzona jest nowa tabela z dokładną datą oraz godziną, która jest zawarta w nazwie tabeli.

Metoda `insert_values_localization()` odpowiada za wstawianie do oddzielnej tabeli danych połączonych ze sobą kluczami głównymi wartości geograficznych potrzebnych do późniejszej analizy danych.

Metoda `save_values_measurement_to_json_file()` odpowiada za tworzenie pliku JSON (ang. JavaScript Object Notation) który jest otwartym formatem zapisu struktur danych. Jego przeznaczeniem jest najczęściej wymiana danych pomiędzy aplikacjami. JSON składa się z par atrybut – wartość oraz typów danych tablicowych. Notacja JSONa jest zbieżna z obiektami w języku JavaScript.

Najważniejszymi elementami kodu programu w głównym bloku programu jest inicjacja wcześniej odpowiednio przygotowanych klas z ich odpowiednimi metodami. Do obsługi czytnika SDS018 została zaimplementowana klasa o nazwie `sds_reader`, która w konstruktorze zawiera odpowiednie parametry dzięki, którym urządzenie może połączyć się z odpowiednim portem. Konstruktor klasy korzysta z pakietu `pyserial`, który pozwala na odczyt z portu szeregowego.

```
import serial

class sds_reader:
    def __init__(self):
self.ser = serial.Serial('/dev/ttyS0', baudrate=9600)
```

Listing 2. Klasa `sds_reader`

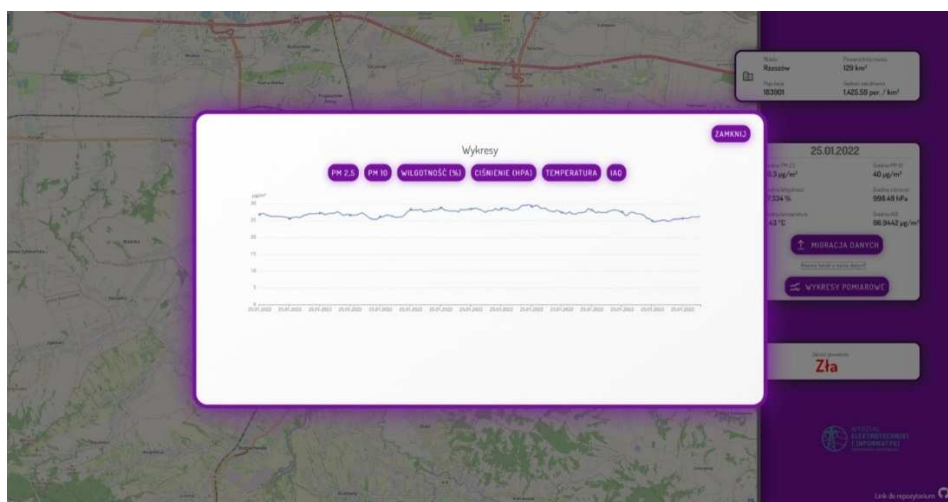
Pierwszym parametrem podawany przez użytkownika jest nazwa urządzenia np. / Dev / ttyUSB0 w systemie GNU / Linux lub COM3 w systemie Windows.

Drugi parametrem jest baudrate który jest typem szybkości transmisji: int domyślnie: 9600 wartości standardowe: 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200. W przypadku chęci odczytania określonej liczby bajtów z urządzenia szeregowego.

Również w przypadku połączenia z czujnikiem BME680 wykorzystywany jest pakiet pyserial do komunikacji z szeregową pomiędzy mikrokomputerem a czujnikiem.

3. APLIKACJA WEBOWA

Jako dodatkowy element ułatwiający segregację i konteneryzację danych została stworzona aplikacja internetowa aby informować o poziomie zanieczyszczenia w badanych miejscach. Aplikacja została napisana w frameworku Angular.



Rysunek 2. Wizualizacja zebranych danych w aplikacji webowej

Źródło: opracowanie własne.

Przedstawiony wygląd zewnętrzny dla użytkownika może wyświetlać poszczególne wykresy pyłów PM_{2,5} PM₁₀, wilgotności, ciśnienia, temperatury oraz obliczonego współczynnika wartości danych IAQ (IndoorAirQuality). Wykresy przedstawiają dokładną datę wraz z godziną i dokładnymi minutami pomiarów. Aplikacja w przyszłych aktualizacjach będzie wzbogacona o możliwość śledzenia trasy, po której przelatywał bezzałogowy statek powietrzny. Na mapie będą widoczne automatycznie pokolorowane punkty, w których widoczne będzie dla użytkownika zanieczyszczenie powietrza nad daną lokalizacją. Kolorystyka automatycznie generowanych kropek będzie przyjazna i przejrzysta dla końcowego

użytkownika. Czerwień będzie oznaczać kolor, w którym zanieczyszczenie jest bardzo duże aż po kolor zielony, który będzie oznaczał, że pomiary w danym miejscu są zgodne z normami środowiskowymi. Na głównej widocznej po prawej stronie panelu widoczna jest lokalizacja, w której był wykonywany pomiar wraz z poszczególnymi danymi dotyczącymi lokalizacji (jej powierzchnia, nazwa miejscowości, powierzchnia wyrażona w kilometrach kwadratowych, oraz zagęszczenie ludności). Pod panelem głównym widoczny jest ogólny wskaźnik czystości powietrza, który wskazuje uśrednione wartości dla danych pomiarów.

4. LISTA KONTROLNA

W celu usprawnienia procesu pomiarowego została wykonana lista kontrolna używana przed każdym rozpoczętym startem:

- Sprawdzenie aktywności stref powietrznych,
- Uzyskanie pozwolenia w danej strefie powietrznej (jeśli wymagane),
- Sprawdzenie ogólnego stanu statku powietrznego,
- Sprawdzenie śmigieł,
- Sprawdzenie poziomu naładowania akumulatora,
- Montaż akumulatora,
- Sprawdzenie poziomu naładowania akumulatora aparatury,
- Sprawdzenie poziomu naładowania akumulatora komputera,
- Sprawdzenie poprawności działania elementów pomiarowych,
- Włożenie karty SD do elementów pomiarowych,
- Sprawdzenie sygnału GPS,
- Podłączenie zasilania,
- Sprawdzenie aktualizacji oprogramowania,
- Sprawdzenie połączenia poziomu sygnału aparatury z dronem,
- Sprawdzenie poprawności działania telemetrii,
- Planowanie trasy przelotu,
- Uzbrojenie drona.

5. PODSUMOWANIE

Bezzałogowy statek powietrzny opisany w artykule idealnie nadaje się do prowadzenia badań w celach sprawdzania jakości powietrza w wybranych lokalizacjach. Podczas tworzenia projektu zostały stworzone także wskazówki eksploatacyjne potrzebne do poprawnego użytkowania bezzałogowego statku powietrznego.

1. Akumulatory należy przechowywać oraz ładować w specjalnych ognioodpornych opakowaniach. Należy pilnować poziomu naładowania akumulatorów podczas przechowywania (zalecane napięcie 3,8V na celę).

2. Należy przechowywać drona w suchym miejscu, unikać kontaktu z wodą.
3. Należy regularnie sprawdzać stan techniczny drona.
4. Przed zainstalowaniem nowej aktualizacji należy zapoznać się co nowego pojawi się w urządzeniu.
5. Loty należy wykonywać w miejscach nienarażonych na silne zakłócenia elektromagnetyczne (linie wysokiego napięcia).
6. Przed lotem należy zapoznać się z warunkami eksploatacyjnymi. Szczególnie ważnym warunkiem jest aktywność magnetyczna Słońca.
7. Operator bezzałogowego statku powietrznego musi posiadać świadectwo kwalifikacji (UAVO).

Wskazówki eksploatacyjne bezzałogowego statku powietrznego zostały opracowane przed tworzeniem procesu projektowania i tworzenia bezzałogowego statku powietrznego, dzięki czemu w trakcie wykonywania pomiarów smogowych dysponowaliśmy odpowiednią wiedzą na temat eksploatacji bezzałogowych statków powietrznych.

Reasumując cały proces tworzenia systemu do badania jakości powietrza przy użyciu bezzałogowego statku powietrznego objął wiele dziedzin, poprzez projektowanie i druk 3D, aż po programowanie w językach Python, Angular.ts, Javascript i naukę prototypowania aplikacji pod względem użyteczności i prostoty dla końcowego użytkownika.

LITERATURA

1. Erich Gamma, Richard Helm, Ralph Johnson Wzorcowe projektowanie. Elementy oprogramowania obiektowego wielokrotnego użytku Wydawnictwa Helion 2020.
2. Krykowski Krzysztof, *Silniki PM BLDC: właściwości, sterowanie, aplikacje*, Wydawnictwo BTC, Legionowo 2015.
3. Halliday Brian, *Drones: The Professional Drone Pilot's Manual*, Wydawnictwa CreateSpace Independent Publishing Platform, 2016.
4. Halliday Brian, *Drones: Mastering Flight Techniques*, Wydawnictwa CreateSpace Independent Publishing Platform, 2017.
5. Ballaster Rufus, Firman Andrew, Clot Eleanor, *A practical Guide to Drone Law*, Wydawnictwo Law Brief Publishing Ltd, 2017.

ŹRÓDŁA INTERNETOWE

6. https://www.alldatasheet.com/view.jsp?Searchword=Esp32%20datasheet&gclid=Cj0KCQjw0umSBhDrARIsAH7FCoc3w_1KUFSGhko7esmi2UJHd_nUchYeGIUIUeyX9355ZYySGC0Y_BlaAl6wEALw_wcB
7. <http://radiolink.com.cn/doce/UploadFile/ProductFile/PIXHAWKManual.pdf>
8. https://www.raspberrypi.com/documentation/computers/linux_kernel.html

Rafał NAZARKO, Piotr STOREK, Karol SIWIEC, Marek GIŻA

dr inż. Bartosz PAWŁOWICZ
opiekun naukowy

SYSTEM IDENTYFIKACJI ZAWODNIKÓW ORAZ DOKONYWANIA POMIARÓW CZASU TRWANIA PRZEJAZDÓW NA ZAWODY ROBO~MOTION

Artykuł opisuje sposób wykonania i zasadę działania systemu do identyfikacji Zawodników podczas zawodów Robomotion. W skład systemu identyfikacji zawodników wchodzi aplikacja desktopowa napisana w języku C++ przy użyciu narzędzi widżetowych do tworzenia interfejsów graficznych oraz aplikacji wieloplatformowych, które działają na różnych platformach programowych i sprzętowych, takich jak Linux, Windows, macOS, Android. Jako układ, który pozwolił na połączenie aplikacji desktopowej z architekturą sprzętową posłużyło ESP32 wraz z komunikacją poprzez UART. Układ charakteryzuje się on większą mocą obliczeniową wynikającą z użycia 2-rdzeniowego procesora taktowanego z częstotliwością do 240 MHz. ESP32 jest połączone z modułem czytnika RFID NFC PN532 pracujący w częstotliwości 13,56MHz. Elementem, który dodatkowo był włączony w badanie pomiarów czasu trwania przejazdów były czujniki przerwania wiązki laserowej, które natychmiast po przejechaniu dowolnego robota przez linię startu bądź mety wychwytywały przerwanie i systemowo kończyły lub zaczynały odmierzać czas przejazdu danego robota. System identyfikacji zawodników świetnie sprawdził się podczas ubiegłej edycji zawodów Robomotion 2021, dzięki czemu usprawnił proces rejestracji zawodników.

Słowa kluczowe: aplikacja desktopowa, Qt5framework, ESP32, RFID, NFC, C++.

WPROWADZENIE

Zainteresowanie technologią RFID (Radio-Frequency Identification) ciągle rośnie, zwłaszcza w branżach logistycznych i zaopatrzeniowych. Technologia ta identyfikuje obiekty na podstawie działania fal radiowych. Informacje zebrane z różnych kart bądź znaczników RFID zapisywane są w specjalnych chipach, które przymocowuje się do przedmiotów, kart lub osób. Dzięki zastosowaniu technologii możliwe jest przesyłanie i odczytywanie danych w odległości nawet

kilkunastu metrów. RFID cieszy się szczególnym uznaniem w procesach przemysłowych i produkcyjnych, automatyzuje procesy oraz identyfikuje poszczególne etapy i zwiększa efektywność dzięki przesyłaniu danych do analizy w czasie rzeczywistym. W logistyce pozwala nadzorować rotację towarów, dostawy oraz wysyłki. W budownictwie bardzo popularne staje się wykorzystanie technologii w przypadku kontroli narzędzi i prowadzeniu ewidencji elementów. Bardzo popularne są także znaczniki wykorzystywane w handlu, dzięki którym zapewniona jest ochrona przed kradzieżą. Głównymi korzyściami z zastosowania technologii RFID to znaczące przyspieszenie identyfikacji obiektów i przedmiotów, tagi RFID są zazwyczaj niewielkich rozmiarów dzięki czemu można przytwierdzić je na małej powierzchni. Kolejnym istotnym plusem jest szybki odczyt danych, który oszczędzi czas co przekłada się na lepszą wydajność pracy i jej efektywność. Technologia RFID doceniana jest również za szerokie możliwości zastosowania, trwałość elementów systemu oraz wygodę jego użytkowania. Coraz więcej firm korzysta z tych rozwiązań, starając się zautomatyzować i usprawnić zachodzące w nich procesy.

Głównymi zadaniami stworzonego systemu było kolejno: przypisywanie nowego zawodnika do karty RFID, Identyfikacja zawodnika oraz pomiar czasu trwania przejazdu. W poszczególnych punktach został opisany zakładany przebieg poszczególnych procesów, które zostały użyte przed lub w trakcie identyfikacji zawodników. Dodatkowo ważną cechą przy tworzeniu całego systemu była łatwa skalowalność i możliwość przenoszenia aplikacji na różne platformy sprzętowe. Głównym celem przy tworzeniu aplikacji w przypadku pojęcia UI/UX (Projekt interfejsu użytkownika oraz doświadczenie użytkownika przy korzystaniu z aplikacji) był minimalizm i prostota. Aplikacja składa się z trzech możliwych do kliknięcia okien, które odpowiednio przenoszą do pozycji „Odczyt”, „Nowy UID” oraz „Odczyt czasu z bramek”. Nazewnictwo przy tworzeniu specjalnych zakładek zostało dobrane w sposób, dzięki któremu każdy użytkownik, sędzia bądź administrator będzie wiedział bez przechodzenia specjalistycznego szkolenia, w którym miejscu należy wybrać przycisk, aby zrobić odpowiednią rzecz. Zapisywanie nowego tagu UID jest możliwe po podaniu specjalnego kodu serwisowego zapisanego bezpośrednio w kodzie aplikacji frontendowej. W zakładce „Odczyt” musimy nawiązać połączenie z mikrokomputerem ESP32, w celach sprawdzenia poprawności nawiązania połączenia zostały umieszczone dwa przyciski o nazwie „LED off” oraz „LED on”, które bezpośrednio i szybko informują administratora o możliwych kolizjach, które związane były z błędami połączeniowymi. Dodatkowym ułatwieniem dla administratorów jest możliwość skopiowania ostatnio przeczytanego tagu UID bezpośrednio do schowka w komputerze bądź laptopie.

Zakładka „Nowy UID” przeznaczona jest dla administratorów systemu, którzy odpowiedzialnie mogą przypisać nowy tag UID zespołowi, a zakładka „Odczyt czasu z bramek” pozwala na wybór odpowiedniej bramki pomiarowej oraz skopiowanie czasu przejazdu do schowka.

1. PRZYPISYWANIE NOWEGO ZAWODNIKA DO KARTY RFID

Jedną z najważniejszych zakładanych funkcjonalności aplikacji jest możliwość przypisania do danej karty RFID unikalnego tagu UUID poprzez odrębną zakładkę w aplikacji.

Zakładany przebieg przypisywania tagu UUID do karty RFID:

- uruchomienie aplikacji wykonywalnej,
- wybranie odpowiedniego portu, do którego podłączone jest ESP32 wraz z czytnikiem RFID,
- aktywowanie trybu zapisu do karty w zakładce Nowy UUID,
- wpisanie odpowiedniego tagu UUID w aplikacji i kliknięcie przycisku zapisz,
- przyłożenie identyfikatora z zaimplementowaną kartą RFID do czytnika RFID.

2. IDENTYFIKACJA ZAWODNIKA

Założeniem systemu identyfikacji zawodnika było szybkie i bezbłędne zidentyfikowanie danego użytkownika oraz robota, którego jest konstruktorem. W tym celu każdy użytkownik otrzymywał identyfikator zawodnika z zaimplementowaną kartą RFID do której w fazie przygotowywania identyfikatorów został przypisany unikalny tag UUID.

Zakładany przebieg identyfikacji:

- uruchomienie aplikacji wykonywalnej,
- wybranie odpowiedniego portu, do którego podłączone jest ESP32 wraz z czytnikiem RFID,
- aktywowanie trybu odczytu karty RFID,
- przyłożenie identyfikatora z zaimplementowaną kartą RFID do czytnika RFID,
- skopiowanie tagu UUID do schowka przy pomocy przycisku znajdującego się w aplikacji,
- wklejenie otrzymanego tagu UUID do bazy danych poprzez aplikację webową, która na podstawie UUID wyświetla dane zawodnika i jego robotów zapisane w bazie danych.

3. POMIAR CZASU TRWANIA PRZEJAZDU

Po udanej identyfikacji zawodnika, mamy możliwość skorzystania z kolejnej funkcjonalności aplikacji jaką jest odczytywanie czasu przejazdu na podstawie wskazań przerwań wiązki światła podczerwonego bramek pomiarowych. Pomiar

czasu trwania przejazdu jest zależny od wymagań danej trasy, a mianowicie aplikacja może mierzyć czas przejazdu przy pomocy jednej lub dwóch bramek pomiarowych. W przypadku wybrania opcji z jedną bramką pomiarową czas przejazdu liczony jest od momentu wykrycia pierwszego przerwania wiązki do momentu wykrycia kolejnego przerwania wiązki, które może nastąpić nie wcześniej niż po upływie 2 sekund. W przypadku wybrania opcji z wykorzystaniem dwóch bramek czas przejazdu liczony jest od momentu wykrycia przerwania na bramce startowej do momentu wykrycia przerwania na bramce końcowej, które może nastąpić nie wcześniej niż po upływie 2 sekund. Opóźnienie 2 sekund jest istotne w celu wyeliminowania błędnych pomiarów. Niezależnie od wybranej opcji, Sędzia z poziomu aplikacji może w dowolnej chwili anulować rozpoczęty pomiar czasu trwania przejazdu.

Zakładany przebieg czasu trwania przejazdu:

- uruchomienie aplikacji wykonywalnej,
- wybranie odpowiedniego portu, do którego podłączone są bramki pomiarowe,
- aktywowanie odpowiedniego trybu pomiaru czasu trwania przejazdu,
- wykrycie momentu pierwszego przerwania wiązki światła podczerwonego i rozpoczęcie odliczania czasu przejazdu,
- wykrycie momentu drugiego przerwania wiązki światła podczerwonego i zatrzymanie odliczania czasu,
- wyświetlenie otrzymanego czasu przejazdu,
- skopiowanie otrzymanego wyniku do schowka przy pomocy przycisku znajdującego się w aplikacji,
- wpisanie do bazy danych czasu trwania przejazdu w trakcie danego podejścia.

Proces pomiaru czasu trwania przejazdu może również zakończyć się w momencie, gdy sędzia zauważy, że urządzenia wystartowały niepoprawnie lub zawodnik z różnych przyczyn chce powtórzyć przejazd. Aplikacja przystosowana jest do przerwania poprzedniego żądania poprzez ponowne kliknięcie przycisku wysyłania danej komendy.

4. BUDOWA SYSTEMU IDENTYFIKACJI ZAWODNIKÓW

Do projektu wybrany został ESP-WROOM-32 z napięciem zasilania: 5 V – z microUSB, które posiada wbudowany układ 802.11BGN HT40 z 30 wyprowadzeniami GPIO w tym: 3x UART, 3x SPI, 2x I2C, 12-kanałowy przetwornik ADC, 2-kanałowy przetwornik DAC, 2x wyjścia PWM oraz interfejs kart SD.

Do odczytu tagów RFID użyty został czytnik NFC bazujący na układzie PN532 od Elhouse. Obsługiwane karty to: Mifare 1k, 4k, Ultralight, DesFire, ISO/IEC 14443-4 (CD97BX, CD light, Desfire, P5CN072), Innovision Jewel (IRT5001), FeliCa (RCS_860, RCS_854). Do połączenia esp32 z czytnikiem kart

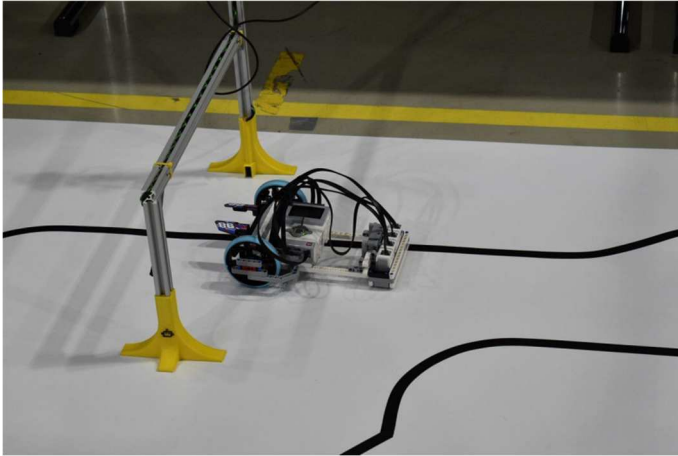
wykorzystana została magistrala SPI, która wykorzystuje poszczególne piny: VCC – podłączone do pinu 5V, GND podłączone do GND, SCK podłączone do GPIO14, MOSI - podłączone do GPIO13, MISO podłączone do GPIO12, SS – podłączone do GPIO04. Wybrany protokół SPI to protokół komunikacji o konfiguracji typu „pełny duplex”. Do komunikacji między urządzeniem nadrzędnym a urządzeniem nadrzędnym wykorzystywane są cztery sygnały: Chip Select (CS), Clock (SCK), Master Out / Slave In (MOSI) oraz Master In / Slave Out (MISO). Pojedyncze połączenie urządzenia nadrzędnego z urządzeniem podrzędnym wymaga jednego sygnału CS. To oznacza, że jeśli do tej samej magistrali zostanie podłączonych więcej niż jedno urządzenie podrzędne, liczba sygnałów CS wzrośnie.



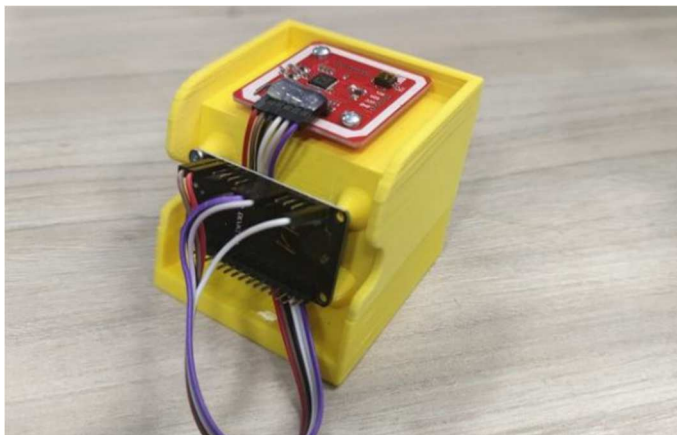
Rysunek 1. Projekt wykonany w technologii druku 3D bramki pomiarowej
Źródło: opracowanie własne.

Do projektu został dobrany także czujnik przerywania wiązki IR – LED 3 mm - 0-50 cm. Elementy podtrzymujące całą konstrukcję były zaprojektowane i wydrukowane przy użyciu technologii druku 3D. Połączenie aplikacji desktopowej z modułem ESP32 odbywało się po porcie szeregowym i magistrali UART. Elementy mocujące wydrukowane w technologii druku 3D posłużyły do mocowania profili aluminiowych. Profile aluminiowe V-Slot są odpowiednikiem tych standardowych profili jednak posiadają jeszcze jedną bardzo ważną funkcjonalność.

Rowki w profilach typu “V”, są specjalnie zaprojektowane w ten sposób aby umożliwić ruchu liniowy. Oprócz klasycznej bramki zostały wykonane również czytniki RFID przeznaczone tylko i wyłącznie do rejestracji zawodników i potwierdzenia ich przybycia. Cały model potrzebował tylko moduł czytnika RFID oraz ESP32. Całość umieszczona była w specjalnie przygotowanym i wydrukowanym elemencie mocującym te dwa elementy.



Rysunek 2. Rzeczywisty model bramki pomiarowej
Źródło: archiwum zdjęć ROBO~motion.



Rysunek 3. Wygląd gotowego czytnika wykorzystywanego przy rejestracji zawodników
Źródło: archiwum własne.

5. KOD ESP32

Kod programu jest podzielony na dwie części. Widok frontendowy czyli widoczna część aplikacji wraz z kodem napisany w języku C++ do obsługi komunikatów przychodzących od mikrokontrolera ESP32, oraz na back-end naszej aplikacji czyli bezpośrednio kod wgrany na płytkę ESP32, który jest mózgiem działania całego systemu. Aplikacja desktopowa napisana w języku C++ obsługuje żądania przychodzące i zapewnia komunikację dwustronną pomiędzy aplikacją desktopową a kodem wgranym i skompilowanym do mikrokontrolera esp32. które odpowiadają za poszczególne elementy identyfikacji zawodników oraz pomiaru czasu.

Poniżej przedstawiony został kod, który wykonywany jest cyklicznie i on sprawdza czy poprzez bufor UARTU został przesłany poszczególne typ reprezentujący pojedynczy znak ASCII. Od znaku, który został przesłany poprzez bufor zależy to która funkcja zostanie wywołana. Poszczególne przyciski na aplikacji desktopowej wysyłają znaki ascii dzięki czemu wiemy, która funkcja zostanie wywołana. Przesłanie do aplikacji 0 lub 1 nadpisuje stan wewnętrznej diody LED, która w celu debugowania programu służy do określenia stanu komunikacji poprzez UART. Wysłanie do mikrokomputera ``2`` wywołują funkcję odczytu tagu RFID, który został wcześniej zapisany na kartę. ``3`` przerywa działanie każdej funkcji w przypadku gdy użytkownik pomylił się i przycisnął niepoprawny przycisk. Przesłanie ``4`` zapisanej w kodzie ASCII zapisuje poszczególne tag UID na kartę zawodnika. Tagi UID zostały wygenerowane w momencie zapisywania użytkownika na zawody ROBO-motion. Przesłanie ``5`` i ``6`` powodują rozpoczęcie odczytu z bramek pomiarowych.

Dyrektywa `#define GATE_1` albo `#define GATES_2` pozwalała po odpowiednim za komentowaniu kodu wgrać na płytkę ESP32 odpowiedni kod w zależności od rodzaju bramki. Funkcja `readGate()` służyła do załączenia systemowej implementacji timera, dzięki któremu na ekranie konsoli po dwukrotnym przekroczeniu bramki poprzez robota został zwracany czas przejazdu. Funkcję `readGate()` można przerwać w każdej chwili poprzez odpowiednie wciśnięcie przycisku w aplikacji desktopowej dla administratorów bądź sędziów. Dzięki zastosowaniu funkcji `millis()`, która zwraca liczbę milisekund, które upłynęły od rozpoczęcia przez płytkę ESP wykonywania bieżącego programu. Liczba ta przepełni się (wróci do zera) po około 50 dniach możemy bardzo dokładnie odliczać czas od wystąpienia pierwszego przerwania do drugiego, które oznajmia, że użytkownik zakończył przejazd.

```
void serialEvent() {
  while (Serial.available()) {
    char state = (char)Serial.read();
    switch(state){
      case '0':
digitalWrite(ledPin, 1);
        break;
      case '1':
digitalWrite(ledPin, 0);
        break;
      case '2':
        if (nfc_read() == 1){
Serial.println("nfcTimeout");
          break;
        }else
          break;
      case '3':
        break;
      case '4':
fillUid(newuid);
nfcWrite(newuid);
break;
      case '5':
Serial.println("Odczyt z jednej bramki");
readGate();
        break;
#ifdef GATES_2
      case '6':
Serial.println("Odczyt z dwóch bramek");
read2Gate();
        break;
#endif
    }
Serial.println("Potwierdzamodbior. Status = " + String(state));
  }
}
```

Listing 1. Szablon funkcji wewnętrznej SerialEvent(), która nasłuchuje odpowiednich pojedynczych znaków.

6. ROZWÓJ PROJEKTU ORAZ NAPRAWIENIE BŁĘDÓW

Opisany w artykule system do identyfikacji zawodników można rozwinąć na wiele sposobów. W przyszłych edycjach zawodów planowane jest między innymi naprawienie drobnych błędów, które wyszły podczas zawodów. Między innymi należy opracować możliwość mechanicznego zresetowania płytki, często w momencie, w którym bufor zostawał przepełniony poprzez zbyt szybkie wysyłanie poszczególnych komend mikrokontroler wraz z czytnikiem RFD zawieszał się. Brak możliwości szybkiego zresetowania mikrokontrolera powodował niezadowolone wśród administratorów zawodów.

Dodatkowym elementem, który jest niezbędny do wdrożenia jest poprawa działania przycisku aplikacji „Skopiuj czas przejazdu do schowka” który do celowo zwracał wynik w sekundach. Trzeba dopracować to i zmienić przedstawiony wynik na milisekundy dzięki czemu zwiększymy dokładność pokazywanych wyników przejazdów robotów. Kolejnym elementem, który wymaga zmiany w przyszłych wersjach aktualizacji jest sposób połączenia ESP32 z czytnikiem RFID. W aktualnej wersji systemu do połączenia służyły tradycyjne przewody połączeniowe z zestawów do szybkiego prototypowania urządzeń elektronicznych. Były to połączenia nietrwałe, łatwe do wypięcia ze względu na ich główne zastosowanie. Przewody te w kolejnych aktualizacjach zostaną wymienione na przewody silikonowe, które bezpośrednio będą przytwierdzone do pinów ESP przy użyciu metody trwałego łączenia elementów metalowych za pomocą metalowego spoiwa zwanego lutem o temperaturze topnienia niższej niż temperatura topnienia łączonych elementów.

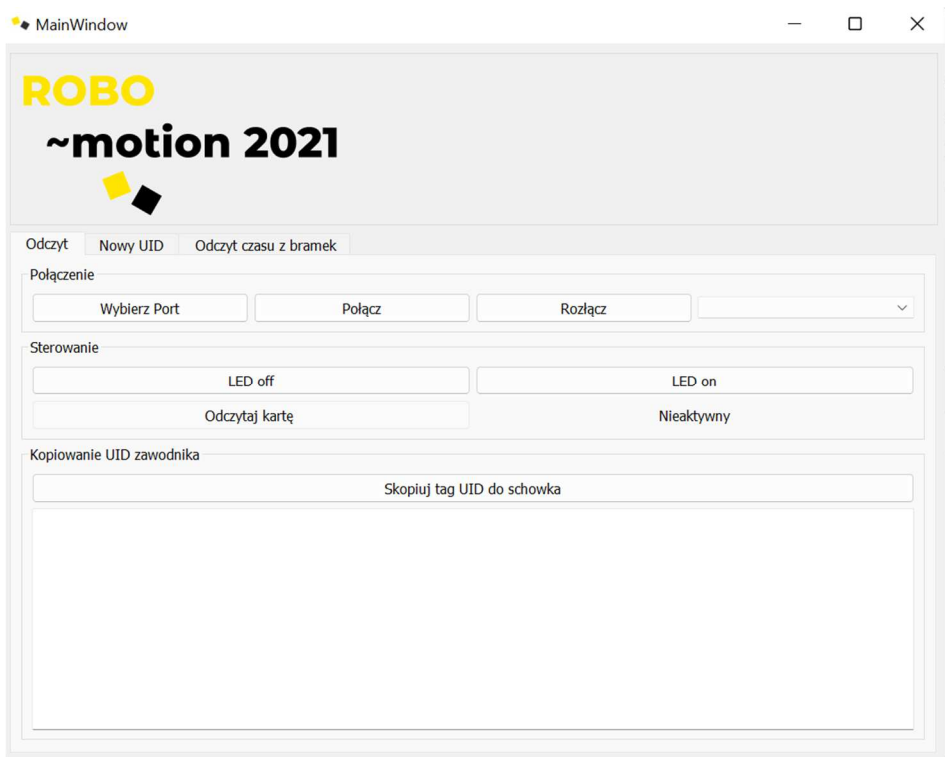
Kolejną nową cechą systemu, która jest planowana do wdrożenia to sytuacja w której aplikacja po odczytaniu tagu RFID w przypadku rejestracji zawodników automatycznie wyśle komunikat do interfejsu programu aplikacyjnego, który w bazie danych odznaczy iż użytkownik stał się na miejsce wraz ze swoim robotem i zgodnie zarówno jak i uczestnik tak i robot pozytywnie przeszli rejestrację.

Elementy wydrukowane w technologii druku 3D również wymagają minimalistycznych poprawek, mianowicie sam czytnik wykorzystywany do potwierdzenia przybycia uczestników zawodów potrzebuje zmiany sposobu mocowania ESP32. Przycisk „reset” do zresetowania pamięci mikrokontrolera w pierwszej wersji znajdują się pod spodem i dostęp do niego jest bardzo utrudniony. Również w przypadku bramek pomiarowych występuje ten sam problem, czyli umiejscowienie przycisku „reset” w trudnym dostępowo miejscu dla administratora.

7. PODSUMOWANIE

System identyfikacji zawodników oraz dokonywania pomiarów czasu trwania przejazdów jest kompletnym systemem do wspomaganie sędziów podczas za-

wodów. Testy systemu przeprowadzone przed zawodami wskazały na drobne niedociągnięcia, które trzeba poprawić w przyszłych wersjach systemu. Konstruktorzy systemu zgodnie stwierdzili, iż w poprzedniej edycji zawodów drobne błędy, które zostały wychwycone w fazie testów nie przeszkadzają do wdrożenia systemu podczas zawodów. Na rysunku nr.4 został przedstawiony wygląd aplikacji co potwierdza przejrzystość i minimalistyczność w zastosowanym systemie. Interfejs użytkownika jest przystosowany do prostych i tradycyjnych kolorów systemu.



Rysunek 4. Zrzut ekranu z pierwszej zakładki aplikacji desktopowej ROBO~motion
Źródło: opracowanie własne.

Na podstawie wykonanych testów wynika jednoznacznie, że przedstawiony system jest możliwy do wykorzystania w przyszłych edycjach zawodów. Reasumując cały proces tworzenia systemu identyfikacji zawodników objął wiele dziedzin, poprzez projektowanie i druk 3D, aż po programowanie w językach C/C++ przy użyciu framework'u QT w wersji 5.5 i naukę prototypowania aplikacji pod względem użyteczności i prostoty dla końcowego użytkownika.

Literatura

1. Erich Gamma, Richard Helm, Ralph Johnson, *Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku* Wydawnictwa Helion
2. Kurniawan Agus, *Internet of Things Projects with ESP3* Wydawnictwa Packt
3. Scott Meyers, *Effective Modern C++* Wydawnictwa O'REILLY' .
4. Nicholas Sheriff, Guillaume Lazar, Robin Penea, Marco Piccolino *End to End GUI Development with Qt5: Develop Cross-platform Applications with Modern UIs Using the Powerful Qt Framework* Wydawnictwo Packt
5. Tomasz Francuz, *Język C dla mikrokontrolerów AVR. Od podstaw do zaawansowanych aplikacji.* Wydawnictwo Helion

Źródła internetowe

6. https://www.alldatasheet.com/view.jsp?Searchword=Esp32%20datasheet&gclid=Cj0KCQjw0umSBhDrARIsAH7FCoc3w_1KUFSgHko7esmi2UJHd_nUchYeGIUIUeyX9355ZYySGC0Y_BlaAl6wEALw_wcB
7. http://www.elechouse.com/elechouse/images/product/PN532_module_V3/PN532_%20ManMan_V3.pdf
8. <https://resources.qt.io/>
9. <https://github.com/piterPR/Aplikacja-QT-Robomotion-2021-QT-ESP32>

Rafał NAZARKO, Piotr STOREK, Karol SIWIEC, Marek GIŻA

dr. inż. Bartosz PAWŁOWICZ
opiekun naukowy

RE-PET ENDER CZYLI MASZYNA DO PRZETWARZANIA BUTELEK PET NA FILAMENT

Artykuł opisuje aspekty ekologiczne, ekonomiczne, sposób działania oraz budowę RE-PET Ender. Maszyna ta ma służyć jako idealny przykład wykorzystania ówczesnych technologii w recyklingu. Sam proces wytwarzania filamentu jest banalnie prosty a materiał wyprodukowany jest twardy, sztywny i odporny chemicznie. Zarządzanie urządzeniem odbywa się poprzez arduino MEGA 3650 z Sterownikiem drukarki 3D reprop. Cel jaki przyświecał nam podczas konfiguracji i budowy maszyny to produkcja filamentu B-Pet z zużytych plastikowych butelek. Artykuł opisuje również z czego stworzona jest nasza nowoczesna maszyna do produkowania filamentu do drukarki 3D oraz sposoby jakie mogą się przyczynić do poprawy jakości tworzonego filamentu. W poniższym artykule znajdziemy również opis wraz z zdjęciami działania naszej maszyny do przetwarzania plastikowych butelek, na rzecz produkcji filamentu do drukarek 3D. Pokazujemy również, jak wygląda nasz pierwszy wydruk zrobiony właśnie z filamentu wykonanego z plastikowych butelek na naszej nowoczesnej maszynie.

Słowa kluczowe: filament, recykling, elektronika, druk 3D, butelka.

WPROWADZENIE

Na przestrzeni lat recykling oraz ochrona środowiska, znacząco wpłynęły na artykuły codziennego użytku. Ogromną popularnością zaczęły cieszyć się ubrania wykonywane z polaru; materiału wykonanego z przetworzonych butelek PET. Recykling sprawia, że jesteśmy w stanie użyć ponownie dany materiał, co sprawia, że ten proces przeprowadzamy nie tylko ze względów ekologicznych, lecz także ze względów ekonomicznych i niespotykanych na rynku kolorów. Plastikowe butelki mogą nam służyć nie tylko do przechowywania różnego rodzaju płynów, mogą zostać również wykorzystane jako materiał do produkcji filamentu do drukarek 3D. Pomysł ten zrodził się w Argentynie, jedna z firm postanowiła wykorzystać materiał jakim jest polietylen zwanym inaczej PET do stworzenia filamentu, materiał ten jest nietoksyczny i łatwo poddaje się recyklingowi. Z roku

na rok rośnie zapotrzebowanie na surowce wynikające z recyklingu PET ze względu na ich zastosowanie w wielu miejscach m.in na meblach, w autach oraz butach. B-Pet czyli materiał który jest ogólnie dostępny w sklepach, produkowany jest właśnie z przetworzonych w odpowiedni sposób plastikowych butelek. Pod tą nazwą kryją się materiał z naprawdę wielkim potencjałem, ponieważ na europejski rynek trafia około 70 miliardów butelek właśnie z tego tworzywa. Materiał ten dostępny jest na rynku tylko w 4 kolorach: transparentnym, białym, czarnym i niebieskim. Technologia ta staje się coraz bardziej popularna na świecie z racji tego, że ludzkość ma coraz większy problem z ilością śmieci jaką produkujemy, oraz z ogromnym wzrostem popularności drukarek 3D, w gospodarstwach domowych. Stworzenie takiej maszyny to idealny przykład recyklingu oraz redukcji zanieczyszczenia środowiska. Materiał ten charakteryzuje się między innymi niewielką chłonnością wody, dobrą odpornością na czynniki atmosferyczne i korozję. PET jest stosunkowo tanim, twardym, sztywnym i odpornym chemicznie materiałem. Re-PET Ender to idealny przykład maszyny, którą udało nam się stworzyć w Kole Naukowym Elektroniki i Technologii Informacyjnych, odpowiadającej na potrzeby i problemy współczesnej cywilizacji bowiem łączy ona 3 najważniejsze aspekty:

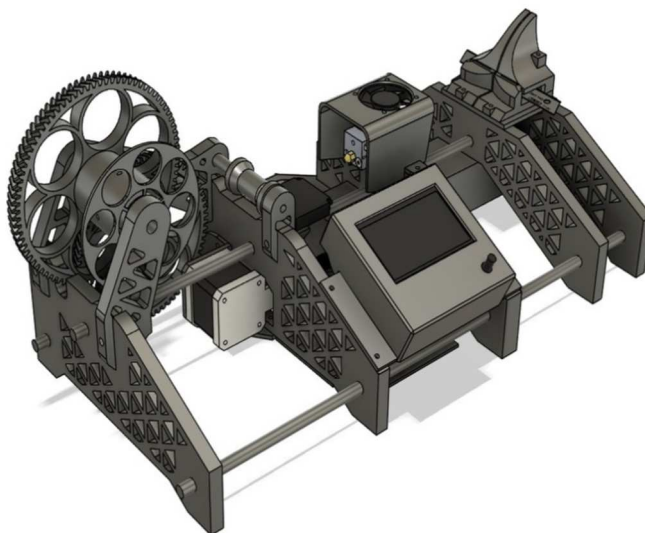
- Redukcja zanieczyszczenia środowiska zużytymi butelkami PET,
- Produkcja filamentu jest zupełnie darmowa,
- Uzyskanie niespotykanych na rynku kolorów.

1. BUDOWA MASZYNY DO PRZETWARZANIA BUTELEK PET NA FILAMENT

Budowę naszej maszyny do przetwarzania butelek PET na filament rozpoczęto od znalezienia projektu open source, który mógłby zostać edytowany pod nasze komponenty. Projekt został wybrany z popularnej wśród środowiska druku 3D platformy Thingiverse¹. Został on przedstawiony na rysunku powyżej. Maszyna składa się z 3 głównych elementów:

1. Od prawej można zauważyć element tnący wraz z uchwytem. Moduł ten składa się z specjalnie zaprojektowanego elementu, który pod wpływem obrotu nacina butelkę tworząc pasek o szerokości od 5 do 12 mm.
2. Na środku znajdują się heatblock z V6 chłodzony wiatrakiem 12V. Jego zadaniem jest przetapianie wcześniej uciętego paska. Również na środku znajdują się ekran LCD z potencjometrem umożliwiającym sterowanie urządzeniem.
3. Po lewej stronie widoczny jest silnik krokowy wraz z zespołem przekładni, które mają na celu naciągnąć powstały filament. Miejscem docelowym, gdzie trafia przetopiony polietylen jest szpulka z największą zębatką.

¹ <https://www.thingiverse.com/>



Rysunek 1. Model maszyny do przetwarzania butelek pet na filament

Źródło: <https://www.thingiverse.com/thing:5255917>.

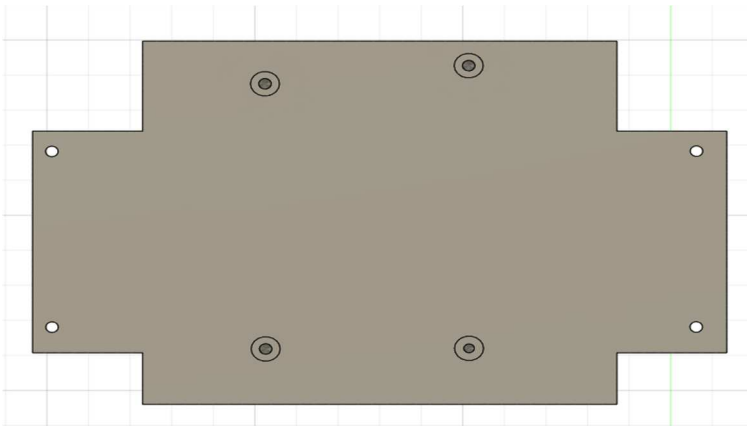
Do budowy maszyny, do przetwarzania butelek PET na filament zostały użyte podzespoły wypisane poniżej:

- silnik krokowy NEMA 17,
- arduino MEGA 2560,
- sterownik drukarki 3D ramps 1.4 RepRap,
- sterownik silnika krokowego A988,
- ekran lcd z potencjometrem,
- hotend J-HEAD V6 RepRap,
- wiatrak 12V,
- turbina Remake 3D 4010 12V,
- zasilacz sieciowy 12V 8 A 96W,
- dysze 0,5 mm,
- łożyska 608zz,
- rolki 625zz,
- wkładki gwintowane M3,
- przetwornica step down,
- wiertło 1,75mm,
- podkładka M6,
- pręt M8,
- nakrętki, podkładki, śrubki M3,M5, M8.

Projekt, który został wybrany jest ogólnie dostępnym modelem 3D opartym na drukarce 3D Ender 3. Projekt został zmodyfikowany w celu dostosowaniu maszyny pod indywidualne potrzeby. Modyfikacje polegały na:

- przeprojektowanie otworów montażowych płyty głównej (Arduino MEGA 2560 z RepRap), oraz dorobienie kanału poprawiającego przepływ powietrza w elektronice
- zaprojektowanie elementu maskującego ekran wraz z potencjometrem (wybrany ekran LCD nieznacznie różnił się od oryginału przez co otwór potencjometru nie pasował),
- Zaprojektowanie mocowania gniazda zasilającego całe urządzenie.

Poniżej przedstawiono przykładowy model, który został przeprojektowany w programie Fusion 360.



Rysunek 2. Przeprojektowany model

Źródło: opracowanie własne.

Arduino MEGA 2560 jest jedną z najmocniejszych płyt z rodziny arduino. W obecnej konfiguracji posiada mikrokontroler ATmega2560 wyposażony w 54 wejścia/wyjścia cyfrowe z czego aż 16 złącz może służyć jako złącza PWM (w naszym przypadku sterowanie silnikiem), oraz 16 analogowych wyjść. Napięcie jakie zasilana jest płyta to w naszym przypadku 8V.

Sterownik drukarki 3D Ramps 1.4 RepRap jest jednym z najpopularniejszych rozwiązań dostępnych na rynku. Cechują się prostotą, dobrym stosunkiem ceny do jakości i umożliwia zarządzanie takimi elementami jak:

- silniki krokowe,
- czujniki krańcowe optyczne i mechaniczne,
- wyświetlacz LCD,
- czytnik kart SD,

- elementy grzejne,
- termistory,
- moduły Bluetooth,
- sterowniki silników krokowych,
- wentylatory,
- pozwala podwoić ilość silników w osi Z.

Jak widać płyta ma ogrom funkcji, jednakże do osiągnięcia zaplanowanego celu, nie była wymagana większość z nich.

Do sterowania silnikiem krokowym NEMA 17 niezbędny jest sterownik silnika A4988, który charakteryzuje się szerokim zakresem napięcia od 8 do 35 V oraz maksymalnym poborze prądu 2A i rozdzielczością 1/16 kroku.

Do zasilenia maszyny użyty został zasilacz 12V, 8A, 96W. Przetwornica step down posłużyła do obniżenia napięcia do 8V, aby zasilić Arduino z sterownikiem drukarki 3D Ramps. Pozostałe elementy elektroniczne użyte w konstrukcji takie jak: grzałka, wentylator chłodzący V6 oraz wentylator chłodzący elektronikę działały na napięciu 12 V.

2. KONFIGURACJA MARLINA

Jako system w maszynie do przetwarzania butelek PET na filament użyty został ogólnie dostępny firmware Marlin. Aby konfiguracja była możliwa należy pobrać środowisko Arduino ze strony producenta

Po konfiguracji środowiska należy otworzyć plik „Marlin.pde” następnie wybierać typ elektroniki, w tym przypadku jest to Ramps 1.4 ”MOTHERBOARD BOARD_RAMPS_14_EFB” (1 ekstruder, stół podgrzewany, wiatrak do chłodzenia wydruku), którą należy skopiować i wkleić w odpowiednie miejsce do pliku „Configuration.h”. Następnie należy zmienić szereg ustawień, aby maszyna działała. Poniżej znajdują się lista z opisami definicji, które należy ustawić:

- #define BAUDRATE 250000; Wartość ta służy do określenia szybkości komunikacji z komputerem,
- #ifndef MOTHERBOARD
- #define MOTHERBOARD BOARD_RAMPS_14_EFB
- #endif ; Rodzaj płyty głównej,
- #define EXTRUDERS 1; Ilość ekstruderów w naszym urządzeniu,
- #define POWER_SUPPLY 1; W tym przypadku zasilacz pochodzi z zestawu zasilającego, taśmę Led dlatego wybieramy 1 w tej opcji,
- #define TEMP_SENSOR_0 1; W maszynie do przetapiania filamentu użyty został tylko jeden termistor, dlatego 1 pojawia się tylko przy pierwszej definicji,
- #define E0_DRIVER_TYPE A4988; Rodzaj sterownika silnika krokowego,

- #define TEMP_SENSOR_BED 0; W tej maszynie nie ma grzanego stołu dlatego należy wyłączyć tę opcję,
 - #define LCD_INFO_SCREEN_STYLE 0; 0 w tym przypadku oznacza klasyczny wygląd interfejsu,
 - #define SPEAKER; opcja pozwala w razie awarii poinformować o problemie poprzez dźwięk,
 - Zmiana kroków silnika na 1020,
Ustawienia LCD oraz karty
 - #define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER; *Rodzaj wyświetlacza,*
 - #define LCD_LANGUAGE en; Język w jakim ma być wyświetlane menu,
 - #define SDSUPPORT; możliwość odczytu z karty SD,
- Ponadto został edytowany szereg innych komend, które usprawniły pracę urządzenia.

3. POZOSTAŁE ELEMENTY

Podczas budowy maszyny do przetwarzania filamentu wykonano szereg prac, między innymi było to rozwiercenie zestawu dysz o średnicy 0,5. Po modyfikacji dysza miała średnicę 1,75mm co odpowiadało standardowej średnicy filamentów dostępnych na rynku. Została dodana przetwornica step down która umożliwiła zasilanie płyty głównej Arduino MEGA 2560 z zasilacza użytego podczas budowy.

4. ZASADA DZIAŁANIA MASZINY DO PRZETWARZANIA BUTELEK PET NA FILAMENT

Do przetworzenia w maszynie nadają się praktycznie wszystkie plastikowe butelki. Przed przetłoczeniem materiału butelkę należy odpowiednio przygotować:

- Pozbyć się wszelakich etykietek,
- Dokładnie umyć butelkę w celu usunięcia kleju.

Przed przetopieniem plastikowej butelki na filament należy ponadto odciąć spód opakowania, ponieważ posiada on bardziej nierównomierne kształty przez co nie nadają się do recyklingu i przetopienia na filament.

Moduł cięcia butelek składa się z elementów, które umożliwiają samoistne nacinanie butelki pod wpływem naciągu z modułu końcowego. Specjalnie zaprojektowany i wydrukowany korpus zawiera w sobie miejsce na montaż noża tnącego który w łatwy sposób nacina butelkę. Ponadto moduł ten posiada regulację szerokości paska od 5 do 12 mm, regulacja ma na celu dobór odpowiedniej sze-

rokości względem grubości ścianek butelki, regulacja oraz dopasowanie odpowiedniej szerokości paska wpływa na optymalizację produkcji filamentu, ponieważ dobór nieodpowiedniej szerokości paska względem grubości ścianki plastikowej butelki może skutkować zatkaniem się dyszy podczas wytwarzania filamentu. Poniżej znajdują się zależności szerokości paska do grubości ścianek butelki:

- 5mm -- > 0,50mm,
- 6mm -- > 0,45mm,
- 7mm -- > 0,40mm,
- 8mm -- >0,35mm,
- 9mm -- > 0,30mm,
- 10mm -- > 0,25mm,
- 11mm -- >0,20mm,
- 12mm -- >0,15mm.

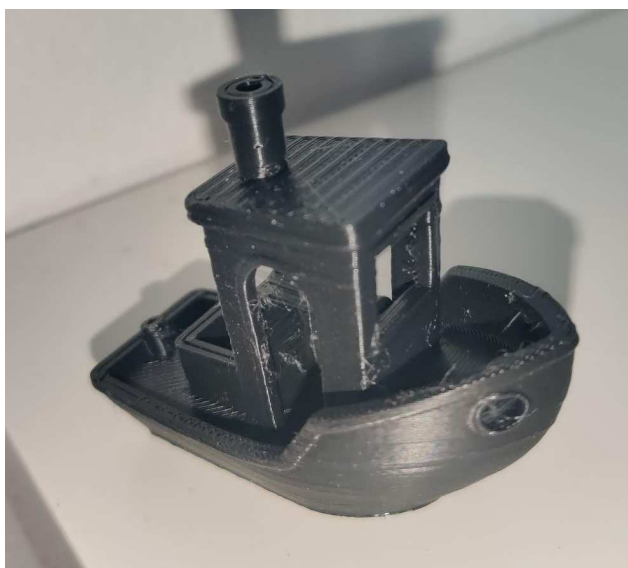
Proces przetapiania polega na doborze odpowiedniej temperatury elementu grzewczego. Średnia temperatura z jaką przetwarzane były plastikowe butelki to 250°C. Jedną z wad maszyny jest fakt, że nie ma dobrego sposobu, aby przeciągnąć pierwsze centymetry paska przez dyszę przez co na początku proces jest stosunkowo problematyczny, pasek wygina się i blokuje w początkowym fragmencie dyszy co powoduje jej zatkanie. Po udanym przetopieniu materiału pierwsze 10 cm filamentu należy przeciągnąć ręcznie z odpowiednią prędkością. Zbyt duża może skutkować zerwaniem się materiału, zbyt mała może powodować zatkanie się dyszy.

Proces nawijania filamentu na szpulę wbrew pozorom składa się z ogromnej ilości części i elementów drukowanych. W skład modułu wchodzi 3 drukowane zębatki, szereg łożysk i elementów usztywniających konstrukcję. Po ręcznym przeciągnięciu około 10 cm filamentu można przystąpić do mocowania początku filamentu na szpuli, która jest miejscem końcowym, gdzie trafia filament. Element ten posiada specjalnie zaprojektowane oczko, przez które należy przełożyć filament, uniemożliwi to późniejsze rozwinięcie się filamentu podczas możliwej awarii lub skończenia się surowca do produkcji. Dobór zębatek oraz silnika krokowego NEMA 17 nie jest przypadkowy, ponieważ odpowiednie przełożenie i kalibracja kroków silnika (finalnie 1020 kroków) skutkowały odpowiednią prędkością tworzenia filamentu. Źle dobrane parametry mogłyby skutkować zbyt szybkim przetapianiem materiału względem dobranej grzałki co mogłoby powodować pogorszenie się jakości i odporności mechaniczno-chemicznej wytworzonego materiału.

5. PIERWSZY WYDRUK Z NASZEGO FILAMENTU

Pierwszym wydrukiem jaki udało nam się wydrukować był popularny wśród środowiska drukarzy 3D test o nazwie „benchy”. Wbrew pozorom ta niewinnie

wyglądająca łódeczka daje ogrom informacji o jakości materiału użytego do druku jak i dokładności samej drukarki. Za materiał posłużyła nam plastikowa butelka po popularnej wodzie gazowanej. Filament wytworzony w Re-PET Ender różni się od filamentów dostępnych na rynku przez co wydruk nie wyszedł idealnie. Dalsze testy pokazały, że wyprodukowany materiał jest bardziej wymagający i wymaga doboru innych parametrów druku. Mimo to po kalibracji i przeprowadzeniu ogromnej ilości testów na różnych modelach 3D uzyskano wyniki, które przerosły nasze oczekiwania. Materiał, który uzyskaliśmy posiadał takie same cechy jak i materiał dostępny na rynku filamentów 3D.



Rysunek 3. Pierwszy wydruk z wykorzystaniem wytworzonego przez nas filamentu
Źródło: <https://www.thingiverse.com/thing:763622>.

6. MOŻLIWOŚĆ ROZWINIĘCIA PROJEKTU

Opisana w artykule maszyna do przetwarzania plastikowych butelek na filament to idealna baza do rozbudowy. Już na etapie wczesnego użytkowania można przypuścić jakie ulepszenia pomogłyby w rozwoju i automatyzacji całego procesu. Poniżej zostały przedstawione przykładowe modyfikacje wraz z opisami:

1. Zmiana sposobu mocowania heat bloku do ramy maszyny, ponieważ w obecnej konfiguracji ciepło zostaje przeniesione na ramę, która zrobiona jest z PETu co powoduje topienie się ramy.

2. Stworzenie dedykowanego oprogramowania sprawiłoby, że maszyna działała by jeszcze sprawniej dzięki dopasowaniu systemu do pełnionej funkcji.
3. Zmiana płyty głównej na lepszą posiadającą sterowniki silnika TMC 2208 lub TMC 2209 polepszyłaby kulturę pracy oraz wydajność.
4. Wymiana elementu grzejnego wraz z heat blokiem na volcano co poprawiłoby skuteczność oraz prędkość przetwarzanego materiału.
5. Przemodelowanie sekcji, w której znajdują się elektronika w celu poprawy kultury pracy, dodanie wentylatora chłodzącego komponenty i poprawa wyglądu samego urządzenia.
6. Dodanie czujnika wykrywającego przerwanie lub koniec filamentu.
7. Poprawa ramy, w celu montażu zasilacza dedykowanego do drukarek 3D, który przyczyniłby się do zwiększenia wydajności urządzenia.
8. Zaprojektowanie elementu nawijającego, który powstały filament nawinie na szpulę, które są używane w drukarkach 3D, ponieważ w obecnej sytuacji aby wydrukować model należy wykręcić cały element wraz z zębatką lub przewinąć filament na zewnętrzną szpulę.
9. Kolejnym sposobem aby poprawić procedurę wytwarzania filamentu mogłoby być połączenie maszyny do przetwarzania butelek PET z autonomicznym ramieniem, które samoczynnie układało by przygotowane wcześniej butelki w określone miejsce, oraz zabierało by nie zdatne do recyklingu górne części butelki zawierające szyjkę oraz gwint.
10. Poprawa aspektów wizualnych naszej maszyny.

7. PODSUMOWANIE

Re-PET Ender, czyli maszyna do przetwarzania plastikowych butelek na filament do drukarek 3D, to urządzenie proste w działaniu jak i obsłudze, lecz niestety posiada wady. Mimo to z powodzeniem da się produkować ze standardowej butelki pet (0,5l) około 10 g filamentu. Aby zapełnić standardową szpulę potrzebujemy więc około 100 butelek.

Na podstawie przeprowadzonych testów można śmiało stwierdzić, że filament który udało się wyprodukować jest równie wytrzymały zarówno chemicznie jaki i mechanicznie. Niczym nie odbiega od produktów powszechnie dostępnych na rynku, co pokazują że taka maszyna może idealnie przyczynić się do redukcji kosztów filamentu, zmniejszając tym samym ilość produkowanych odpadów w postaci plastikowych butelek.

Podsumowując cały proces powstawania maszyny do produkcji filamentu z butelek pet objął szeroki zakres prac między innymi z takich dziedzin jak modelowanie i druk 3d, lutowanie elementów elektrycznych, modyfikację oprogramowania open source Marlin.

ŹRÓDŁA INTERNETOWE

1. <https://www.thingiverse.com/thing:5255917>
2. https://reprap.org/wiki/RAMPS_1.4#Design
3. <https://3dprint.wiki/>
4. <https://ekologistyka24.pl/recykling,ac282/drugie-zycie-butelki-pet-najwazniejsze-informacje,964>
5. <https://marlinfw.org/>
6. <https://github.com>

Rafał NAZARKO, Piotr STOREK, Karol SIWIEC, Marek GIŻA

dr inż. Bartosz PAWŁOWICZ
opiekun naukowy

UNIWERSALNY KOMPUTER POKŁADOWY

Artykuł opisuje zasadę działania minikomputera Raspberry Pi służącego za platformę komunikacyjną z dowolnym samochodem wyposażonym w złącze diagnostyczne OBDII. Zintegrowany system wymiany informacji z samochodem pozwala na odczyt jego parametrów takich jak m.in. prędkość, obroty silnika czy komunikaty o błędach. System automatycznie dopasowuje się do danego samochodu, pozwalając użytkownikowi na korzystanie z pełnego spektrum funkcji udostępnionego poprzez komputer pokładowy.

Słowa kluczowe: programowanie, bezprzewodowość, zintegrowany system.

WPROWADZENIE

Obecnie praktycznie każdy nowy samochód posiada wbudowany panel, pozwalający na odczyt parametrów powiązanych z pracą tegoż samochodu. Służy on również jako urządzenie multimedialne, umożliwiające choćby wybór stacji radiowej, regulację głośności czy nawigację pod wskazany adres. Wszystkie te możliwości ułatwiają kierowcy kontrolę nad prowadzonym pojazdem oraz zwiększają komfort podróży. Na brak takich rozwiązań poskarżyć się mogą posiadacze starszych samochodów lub z uboższą wersją wyposażenia, w których wszelkie multimedia kontrolowane są za pomocą analogowych pokręteł lub przycisków a dostępność danych o parametrach samochodu jest bardzo ograniczona. Naprzeciw takim problemom wychodzą producenci zewnętrzni, którzy usiłują poszerzyć możliwości samochodu o dodatkowe funkcjonalności, oferując gotowe rozwiązania w postaci dedykowanych urządzeń z dotykowym panelem, udostępniającym funkcję kontroli multimediiów, nawigacji itp. Zazwyczaj konkretne urządzenie jest tworzone pod konkretny model pojazdu określonej marki, dlatego główną wadą takiego rozwiązania jest brak uniwersalności w użytku. Na kolejną nieprzychylną uwagę zasługuje fakt, że tego typu urządzenia nie rozwiązują problemu wglądu do aktualnych parametrów pojazdu podczas jego pracy, tym samym ograniczając kierowcy kontrolę nad nim.

W tym artykule zostanie opisane urządzenie, które w założeniu rozwiązuje opisane powyżej problemy. Stworzone ono zostało z powszechnie dostępnych oraz względnie niedrogich komponentów oraz skonfigurowane w taki sposób aby

służyło jako uniwersalny komputer pokładowy. Poprzez lekturę kolejnych rozdziałów tego artykułu czytelnik pozna szczegóły budowy oraz zasady działania tego urządzenia.

1. PLATFORMA URUCHOMIENIOWA

System oparty jest o minikomputer Raspberry Pi 4, w wersji wyposażonej w 4 Gigabajty pamięci RAM oraz architekturę mikroprocesora ARM. Wybór tego komponentu uwarunkowany był wbudowanymi modułami Bluetooth 5.0 oraz Wi-Fi, interfejsami I2C, UART i LCD, bogatym zestawem pinów GPIO a także dobrym wsparciem dla systemu Linux. Złącza microHDMI, USB 3.0 oraz Ethernet pozwalają przekształcić tą małą płytkę elektroniczną w pełnoprawny komputer.

Systemem operacyjnym tej jednostki jest GNU/Linux a dokładnie dystrybucja Ubuntu MATE. Jest on stworzony z myślą o komputerach jedнопłytkowych - dzięki temu jest zoptymalizowany do pracy w systemach wbudowanych tym samym pozwalając osiągnąć lepsze wrażenia z użytkowania. Jest to stabilny, łatwy w użyciu system operacyjny z konfigurowalnym interfejsem graficznym. Dodatkowym ułatwieniem są wbudowane sterowniki do obsługi ekranów dotykowych, dzięki czemu proces rozpoczęcia użytkowania został znacznie uproszczony i skrócony.

Do komunikacji z użytkownikiem używany jest 7 calowy ekran pojemnościowy DSI o rozdzielczości 800x480 pikseli. Urządzenie zostało przemyślane tak, aby po złożeniu i dodaniu obudowy mieściło się w samochodowym slocie 2 DIN w desce rozdzielczej.

2. KOMUNIKACJA Z SAMOCHODEM

Głównym celem projektu jest stworzenie platformy pośredniczącej w komunikacji pomiędzy samochodem a osobami w nim przebywającymi. Aby nawiązać połączenie z samochodem często wykorzystywane jest złącze diagnostyczne OBD oraz OBD-II, które dają możliwość dostępu do parametrów opisujących stany poszczególnych komponentów pojazdu oraz odczytywanie zapisanych w pamięci kodów błędów. System OBD-II jest standardem obowiązującym w samochodach wyprodukowanych po:

- 1 stycznia 1996r w USA,
- 1 stycznia 2001r w Unii Europejskiej (Benzyna) ,
- 1 stycznia 2003r w Unii Europejskiej (Diesel).

Widać zatem, że w taki interfejs jest wyposażona przytłaczająca większość pojazdów aktualnie poruszających się po drogach. Aby jednak dostać się do danych udostępnianych przez samochód należy wyposażyć się w adapter wyposażony w nadajnik bezprzewodowy lub z kablem. Z racji tego, że założeniem tego

projektu jest uniwersalność, bezprzewodowość oraz estetyka, wybór padł na adapter ELM327 komunikujący się poprzez Bluetooth.

Opisany zostanie teraz proces łączenia adaptera z wbudowanym w Raspberry odbiornikiem Bluetooth.

1. W pierwszym kroku należy doprowadzić zasilanie do minikomputera oraz upewnić się, że dioda w adapterze świeci po wpięciu go do gniazda złącza OBD-II.
2. Zanim zaczniesz konfigurację klienta i połączenia należy zmodyfikować jeden z plików odpowiadający za działanie serwisów Bluetooth. Jest to problem występujący jedynie na minikomputerach Raspberry Pi i występuje on prawdopodobnie na połączeniu pomiędzy sterownikami wbudowanego odbiornika Bluetooth a jądrem systemu Linux. Niemniej jednak poniższa modyfikacja pozwala na zażegnanie tych problemów.

```
pi@raspberrypi: ~$ sudo nano /etc/systemd/system/dbus-  
org.bluetooth.service  
  
...  
ExecStart=/usr/lib/Bluetooth/bluetoothd -C  
ExecStartPost=/usr/bin/sdptool add SP  
...
```

Listing 1. Modyfikacja pliku dbus-org.bluetooth.service (fragmenty kodu napisane pogrubioną czcionką zostały dodane do oryginalnego pliku)

3. Następnie w konsoli Linuxa wprowadzić poniższe komendy. Pozwolą one na skonfigurowanie i włączenie klienta Bluetooth oraz rozpoczęcie przeszukiwania widocznych urządzeń w okolicy w celu odnalezienia adaptera.

```
pi@raspberrypi: ~$ bluetoothctl  
[bluetooth]# power on  
[bluetooth]# pairable on  
[bluetooth]# agent on  
[bluetooth]# default-agent  
[bluetooth]# scan on
```

Listing 2. Komendy do konfiguracji klienta Bluetooth w systemie Linux

4. Jeśli poprzednie kroki zostały wykonane poprawnie i nie napotkano błędów to na ekranie konsoli powinny zostać wylistowane widoczne urządzenia Bluetooth wraz z ich identyfikatorami. Należy odnaleźć na liście urządzenie nazwane OBDII lub podobnie i zapisać jego adres MAC, który

ma postać dwunastoznakowego ciągu, podzielonego na 6 równych segmentów, oddzielonych znakiem dwukropka. Schemat zapisu wygląda następująco: XX:XX:XX:XX:XX:XX, gdzie X jest dowolną liczbą szesnastkową.

5. Po zidentyfikowaniu urządzenia należy się z nim połączyć. Aby uwierzytelnić połączenie należy podać kod PIN, który zazwyczaj to 1234 lub 1111. Komenda trust pozwala na zaufanie danemu urządzeniu, tzn. w przyszłości będzie można się z nim łączyć bez podawania kodu PIN.

```
[bluetooth]# pair 00:1D:A5:68:98:8A
[bluetooth]# trust 00:1D:A5:68:98:8A
```

Listing 3. Komendy pozwalające połączyć klienta z adapterem

6. W tym momencie urządzenia powinny już być ze sobą powiązane i być gotowe do wzajemnej komunikacji. Jednakże, aby ułatwić proces nasłuchiwania, powinno się utworzyć kanał, do którego przyporządkowany zostanie adapter podłączony do gniazda OBD-II. W tym momencie od razu powinno zmienić się prawa dostępu do pliku na 666 czyli dostępne do otwarcia i edycji dla wszystkich użytkowników. Umożliwi to korzystanie z bibliotek komunikacyjnych wykorzystywanych w dalszej części tego artykułu.

```
pi@raspberrypi: ~$ sudo rfcomm bind rfcomm99 00:1D:A5:68:98:8A
pi@raspberrypi: ~$ sudo rfcomm connect /dev/rfcomm99
00:1D:A5:68:98:8A 1 & sudo rfcomm release /dev/rfcomm99 & sleep
10 && sudo chmod 666 /dev/rfcomm99
```

Listing 4. Komendy tworzące kanał komunikacyjny z adapterem, w tym przypadku utworzonego pod portem 99 (dostępne wartości od 0 do 99)

Po wykonaniu powyższych kroków można w końcu wysłać pierwszą komendę. W tym celu można otworzyć aplikację pod nazwą **screen**. Trzeba jednak wpiery wiedzieć, jakich danych adapter oczekuje tzn. jaką ramkę danych jest w stanie obsłużyć. Komendy obsługiwane przez OBD-II składają się z czterech liczb szesnastkowych (2 bajty):

- Pierwsze dwie określają usługę/tryb, który ma zostać użyty do wykonania komendy. Istnieje 10 usług diagnostycznych opisanych w najnowszym standardzie OBD-II SAE J1979. Przed 2002 rokiem J1979 określał te

usługi jako „tryby”. Wszystkie zostały zaprezentowane w poniższej tabeli wraz z krótkim opisem.

Tabela 1. Usługa/tryb wykonywania komendy

Usługa / Tryb (szesnastkowy)	Opis
01	Bieżące dane - Monitory I/M i dane na żywo
02	Dane zamrożonej ramki
03	Zapisane diagnostyczne kody usterek
04	Wyczyść/usuń diagnostyczne kody usterek i zapisane wartości
05	Wyniki testów, monitorowanie czujnika tlenu
06	Wyniki testów, monitorowanie innych komponentów/systemów
07	Oczekujące kody usterek diagnostycznych
08	Sterowanie działaniem podzespołu/systemu pokładowego
09	Informacje o pojeździe
0A	Stałe diagnostyczne kody usterek

Źródło: https://en.wikipedia.org/wiki/OBD-II_PIDs#Services/_/Modes

- Kolejne dwie to ustandaryzowane identyfikatory parametrów diagnostyki pokładowej zwanych PID'ami. Wskazują one na dane, które użytkownik chce pozyskać lub jaką komendę chce wykonać. Nie wszystkie komendy muszą być obsługiwane przez każdy samochód więc przed rozpoczęciem pracy należy sprawdzić jakie PID'y są wspierane przez komputer pokładowy w danym samochodzie. Poniżej zostanie zaprezentowana część tabeli zawierającej kody w Usłudze/Trybie nr. 1 – każda z pozycji jest opisana kodem szesnastkowym, decymalnym, krótkim opisem, możliwymi wartościami i jednostkami.

Tabela 2. Wykaz pierwszej części PID'ów z pierwszej Usługi/Trybu

PID (HEX)	PID (DEC)	Opis	Wart. min.	Wart. max.	Jednostka
00	0	Obsługiwane PIDy [01 - 20]. Zwraca PID'y, które są wspierane przez ten komputer pokładowy	-	-	-
01	1	Monitoruj stan od czasu skasowania kodów DTC	-	-	-
02	2	Zamroź kod DTC	-	-	-
03	3	Stan układu paliwowego	-	-	-
04	4	Obliczone obciążenie silnika	0	100	%
05	5	Temperatura płynu chłodzącego silnik	-40	215	°C

06	6	Krótkoterminowa regulacja paliwa — Bank 1	-100	99,2	%
07	7	Długoterminowa regulacja paliwa — Bank 1			
08	8	Krótkoterminowa regulacja paliwa — Bank 2			
09	9	Długoterminowa regulacja paliwa — Bank 2			
0A	10	Ciśnienie paliwa (ciśnienie manometryczne)	0	765	kPa
0B	11	Ciśnienie bezwzględne w kolektorze dolotowym	0	255	kPa
0C	12	Prędkość silnika	0	16,383,75	obr/min
0D	13	Prędkość pojazdu	0	255	km/h
0E	14	Zaliczka czasu	-64	63,5	°
0F	15	Temperatura powietrza wlotowego	-40	215	°C
10	16	Czujnik masowego przepływu powietrza (MAF) natężenie przepływu powietrza	0	655,35	g/s
11	17	Pozycja przepustnicy	0	100	%
12	18	Nakazany status powietrza wtórnego	-	-	-
13	19	Obecne czujniki tlenu (w 2 bankach)	-	-	-
14-1A	20-26	Czujnik tlenu 1-7 A: Napięcie B: Krótkoterminowa regulacja paliwa	0 -100	1,275 99,2	Volt %
1B	27	Sonda lambda 8 A: Napięcie B: Krótkoterminowa regulacja paliwa	0 -100	1,275 99,2	Volt %
1C	28	Standardy OBD, z którymi ten pojazd jest zgodny	1	250	-
1D	29	Obecne czujniki tlenu (w 4 bankach)	-	-	-
1E	30	Stan wejścia pomocniczego	-	-	-
1F	31	Czas pracy od uruchomienia silnika	0	65535	sekundy
...

Źródło: https://en.wikipedia.org/wiki/OBD-II_PIDs#Service_01

Wystarczy teraz wysłać zakodowaną komendę do adaptera i następnie odczekać na odpowiedź wbudowanego komputera pokładowego. Na potrzeby wyjaśnienia schematu odpowiedzi otrzymanej od adaptera przyjmijmy, że wysłaną komendą była 010D czyli zapytanie o aktualną prędkość, wtedy odpowiedź może wyglądać następująco:

48 6B 10 41 0D 00 11

Odpowiedzi podobnie jak komendy przesyłane są w postaci zakodowanych szesnastkowo wartości. To co składa się na poszczególne segmenty odpowiedzi zostało opisane w punktach poniżej:

- Pierwsze 6 znaków (segmenty 1-3) stanowi nagłówek odpowiedzi
- Kolejne 2 znaki (segment 4) pokazują Usługę/Tryb w jakiej wysłano żądanie do adaptera
- Następne 2 znaki (segment 5) odzwierciedlają PID żądania
- Dopiero na kolejnych 2 znakach (segment 6) zakodowana jest wartość rzeczywista odpowiedzi czyli to co potencjalnie interesuje użytkownika
- Ostatnie 2 znaki (segment 7) to bajty kończące odpowiedź tzw. sekwencja STOP

Jak widać z odpowiedzi, w tym przypadku samochód nie poruszał się ponieważ segment 6 był wyzerowany (wartość 00). Jeśli zaś samochód poruszałby się z prędkością przykładowo 30km/h to w tym miejscu pojawiłyby się znaki 1E. Długość odpowiedzi jest zróżnicowana pomiędzy komendami a ich rzeczywista długość jest zakodowana w nagłówku.

3. BACKEND

Mając już ustanowioną działającą komunikację można przejść do kolejnego punktu a mianowicie do serwisu odpowiadającego za nasłuchiwanie aktualnych parametrów samochodu i przekazywania ich do interfejsu użytkownika. Często stosowanym zabiegiem jest stworzenie zintegrowanego interfejsu programistycznego (API), które zwraca odpowiedź na zapytanie HTTP Request. Taki serwis uruchomiony na localhost nie generuje niepotrzebnego ruchu sieciowego a stanowi bramkę pomiędzy warstwą logiczną a warstwą prezentacji w aplikacji. W tym przypadku wybór padł na szalenie popularny w ostatnim czasie język programowania - Python. Tworzenie API w tym języku przy wykorzystaniu zdarzeń asynchronicznych jest bardzo proste i znacznie przyspiesza proces programowania. Dodatkową zaletą używania tego języka jest istniejąca biblioteka Python-OBd, która to pozwala na odczytywanie danych pojazdu bez konieczności pisania komend w postaci szesnastkowej. Wszystkie możliwe komendy zostały spisane przez autora tej biblioteki, pozwalając również na odpytywanie komputera w sposób asynchroniczny tzn. użytkownik dostaje informację wtedy, kiedy zmieni się któryś z listy nasłuchiowanych parametrów samochodu. Inicjacja połączenia z samochodem odbywa się poprzez utworzenie poniższego obiektu. Parametry połączeniowe podawane są jako argumenty konstruktora obiektu `Obd.Async`. Po wykonaniu tej linijki kodu pod zmienną `connection` znajduje się zainicjalizowane połączenie nasłuchiwane na porcie `/dev/rfcomm99`, przy prędkości 9600 symboli na sekundę, z czasem oczekiwania na odpowiedź maksymalnie 10 sekund. Opis pozostałych parametrów dostępny jest w dokumentacji biblioteki.

```
connection = obd.Async(portstr="/dev/rfcomm99", baudrate="9600",
fast=False, timeout="10", check_voltage=False, protocol="6")
```

Listing 5. Inicjacja połączenia API z adapterem

Z racji tego, że zostało zainicjowane połączenie asynchroniczne, należy teraz podać jakie dane powinny zostać dodane do nasłuchiwanie zmian. Jako argumenty funkcji *watch* należy podać nazwę komendy oraz funkcję obsługującą nadejście nowego zdarzenia (nowej wartości) dla tej komendy. Aby mieć pewność, że dana komenda jest obsługiwana przez konkretny komputer pokładowy można sprawdzić jej kompatybilność poprzez funkcję *supports*. Przykładowy sposób użycia został zaprezentowany poniżej. Funkcja *new_async_value* przyjmuje nową wartość otrzymaną od adaptera i przypisuje ją pod odpowiedni klucz w mapie danych:

```
whitelist_commands = [
    obd.commands.RPM,
    obd.commands.SPEED,
    obd.commands.THROTTLE_POS,
    obd.commands.COOLANT_TEMP,
    obd.commands.ELM_VOLTAGE,
    obd.commands.GET_DTC
]

def new_async_value(response):
    # manage new value from OBD2
    if not response.is_null():
        json_data[str(response.command)[9:].strip().replace("
", "_").lower()] = response.value.magnitude if (str(type(response.value)) == "<class 'pint.unit.build_quantity_class.<locals>.Quantity'>") else str(response.value)
        loop = asyncio.new_event_loop()
        asyncio.set_event_loop(loop)
        loop.run_until_complete(send_json_data())

for command in whitelist_commands:
    if connection.supports(command):
        connection.watch(command, callback=new_async_value)
```

Listing 6. Dodanie nasłuchiwanyc zdarzeń

Z poziomu interfejsu należałoby wykonywać cykliczne zapytania do API, nawet jeśli nie pojawiłaby się nowa wartość. Jest to oczywiście zjawiskiem niepożądanym, ponieważ pomimo że nie zwiększa ruchu sieciowego poza urządzeniem to niepotrzebnie zużywa jego zasoby. Z rozwiązaniem wychodzą naprzeciw WebSockets, które działają podobnie jak wspomniane wcześniej zdarzenia asyn-

chroniczne – wysyłają nowe dane, kiedy tylko nadejdą. Klient podłączony do takiego serwera WebSocket będzie w czasie zbliżonym do rzeczywistego informowany o nowych danych bez potrzeby ciągłego odpytywania serwera. Dane zostaną przekazane użytkownikowi w postaci mapy klucz wartość, czyli powszechnie znanego JSON'a. Aby jednak nie narzucać aplikacji obsługującej interfejs zbyt wiele zdarzeń do obsłużenia, częstotliwość rozsyłania nowych danych została tutaj ograniczona do 2 razy na sekundę. Poniżej znajduje się kod odpowiedzialny za wyżej wymienione funkcjonalności:

```
async def send_json_data():
    # send new data to all clients
    global debug_max_inputs
    # firewall to prevent too frequent sending of data
    last_send_time = datetime.strptime(json_data["last_send"],
    '%Y-%m-%d %H:%M:%S.%f')
    time_diff_milis = (datetime.now() - last_send_time).total_seconds() * 1000
    if time_diff_milis > 500: # if last send data is longer than
    a specified delay
        json_data["last_send"] = str(datetime.now())
        for conn in connected: # for every connected client
            await conn.send(str(json_data))
            print("sended: ", json_data)
        if not debug_mode: # save data to debug data
            with open(save_debug_file_path, "r+") as file_object:
                debug_lines_count = len(file_object.readlines())
                if debug_lines_count < debug_max_inputs:
                    temp_json = json_data.copy()
                    temp_json.pop("last_send", None)
                    file_object.write(json.dumps(temp_json) +
                    str('\n'))
```

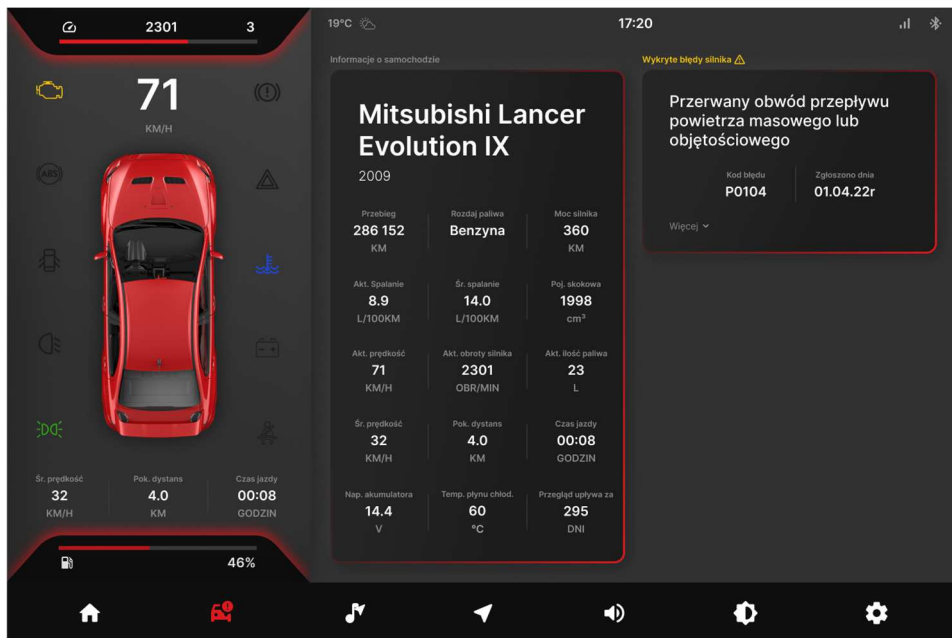
Listing 7. Funkcja obsługująca WebSocketsy

4. FRONTEND

Na chwilę pisania tego artykułu interfejs jest dopiero we wczesnej fazie budowy. Poniżej zostaje zamieszczona wizualizacja jednego z ekranów, który będzie dostępny w ramach aplikacji.

Podczas tworzenia interfejsu wykorzystywana jest nowoczesna technologia Flutter, która pozwala tworzyć aplikacje na praktycznie wszystkie możliwe platformy mobilne, desktopowe a także webowe z poziomu jednego kodu. Tworzone w tej technologii aplikacje cechują się również bardzo wysoką płynnością, zbliżoną do aplikacji napisanych w natywnych językach danej platformy.

Zgodnie z założeniami interfejs ma być przejrzysty i estetyczny, a także pozostawiać użytkownikowi możliwości ustawienia osobistych preferencji co do koloru, języka lub rozmiaru czcionki.



Rysunek 1. Wizualizacja ekranu statusu samochodu

Źródło: opracowanie własne

5. PODSUMOWANIE

Przedstawiony w tym artykule komputer pokładowy jest jeszcze w fazie wczesnej produkcji, ale już w tym momencie są planowane kolejne rozszerzenia i dodatki. Takowa platforma jest bardzo podatna na modyfikację, które można łatwo wprowadzić nawet po zainstalowaniu urządzenia w miejscu przeznaczenia. Jednym z ciekawszych pomysłów na dodatkową funkcjonalność jest dodanie kamery cofania oraz kamery obserwującą kierowcę podczas jazdy. Ta druga miałaby za zadanie rozpoznawanie wyrazu twarzy kierowcy, proponując przykładowo postój, gdy jest poirytowany lub wyszukujący miejsce noclegowe, gdy jest zmęczony. Dodatkowo można wprowadzić asystenta głosowego, który wykonywałby werbalne polecenia kierowcy, tym samym ułatwiając prowadzenie samochodu i obsługę komputera pokładowego.



<https://github.com/Rav-Naz/Car-Computer>

Rysunek 2. Kod QR zawierający adres do repozytorium projektu

Źródło: opracowanie własne

LITERATURA

1. Ceder N., *Python szybko i prosto. Wydanie III*, Wydawnictwo Helion, 2019
2. Gay W., *Advanced Raspberry Pi: Raspbian Linux and GPIO Integration*, Springer Science, 2018

ŹRÓDŁA INTERNETOWE

3. *Biblioteka Python-OBD*, <https://python-obd.readthedocs.io/en/latest/> (dostęp 13.04.2022r)



KOŁO

NAUKOWE

○ ELEKTRONIKÓW



Patryk JASKUŁA, Hubert SKALSKI, Łukasz SZKARADEK, Krzysztof SADOWY

dr inż. Mirosław SOBASZEK
opiekun naukowy

WYŚWIETLACZ PRZEMYSŁOWY OPARTY NA MATRYCACH LED

Artykuł opisuje projekt, wykonanie oraz zasadę działania przemysłowego wyświetlacza opartego na matrycach LED. Rozwiązanie użyte przez nas pozwala na kaskadowe łączenie z sobą matryc LED, dzięki czemu możliwe jest płynne przesuwanie wyświetlanych komunikatów. Wyświetlacz może być częścią innych urządzeń elektronicznych np. pomiarowych i informować o aktualnej wartości pomiaru lub o stanie danego urządzenia do którego wyświetlacz jest podpięty. Wyświetlacz ponadto może posłużyć do wyświetlania: komunikatów, aktualnego czasu oraz parametrów środowiskowych takich jak aktualna temperatura, ciśnienie, wilgotność powietrza a także innych parametrów środowiskowych.

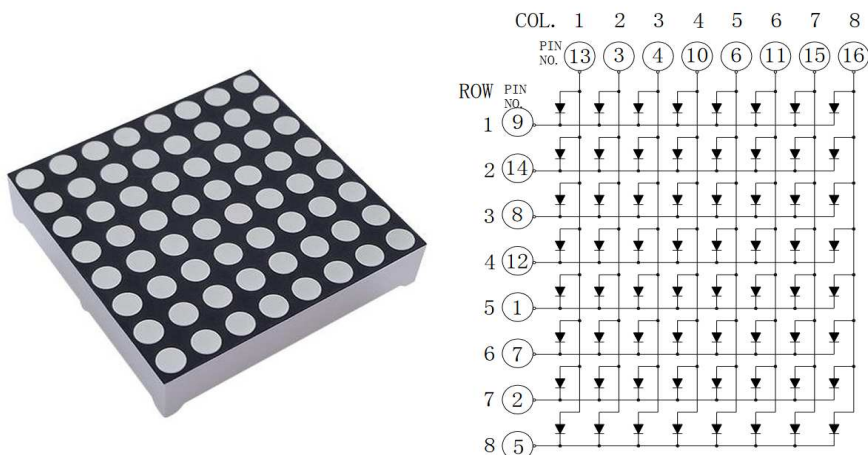
Słowa kluczowe: Wyświetlacz LED, Matryca LED, PIC32MX, SPI, MAX7219.

WPROWADZENIE

Wyświetlacze w elektronice potrzebne są do komunikacji urządzenia z człowiekiem, służą do wyświetlania w zrozumiały dla człowieka sposób pożądaných przez niego informacji lub komunikatów. W naszym projekcie stworzyliśmy wyświetlacz LED 64 x 16, który charakteryzuje się prostą, zwartą konstrukcją i dużymi możliwościami wyświetlania. Wyświetlacz ten w swojej budowie wykorzystuje sterowniki MAX7219 oraz matryce LED 8 x 8 KWM-30881CVB. Do przesyłania informacji z mikrokontrolera do układu MAX7219, które mają zostać wyświetlone na matrycach wykorzystywany się protokół komunikacyjny SPI, transmisja ma charakter Master-Slave i podobnie jak w I2C do magistrali można podpiąć kilka urządzeń typu master oraz slave. SPI nie ma ograniczenia co do ilości urządzeń na jednej magistrali, lecz ich liczba jest uzależniona od liczby dostępnych portów układu master oraz od pojemności pasywnych, które mogą ograniczyć szybkość transmisji danych.

1. BUDOWA WYŚWIETLACZA PRZEMYSŁOWEGO OPARTEGO NA MATRYCACH LED

Projekt budowy wyświetlacza LED 64 x 16 został podzielony na kilka faz związanych z poszczególnymi etapami budowy. Jako roboczy element wyświetlacza wybrana została matryca LED KWM-30881CVB produkcji firmy Lucky-light.



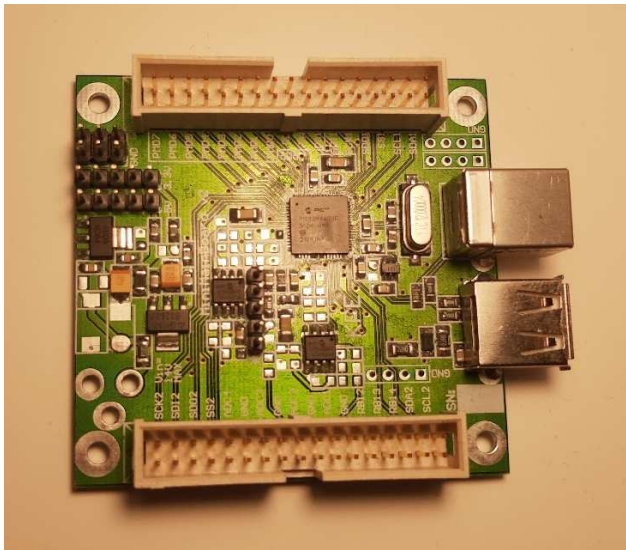
Rys. 1. Matryca KWM-30881CVB produkowana przez firmę Luckylight.

Matryca składa się z 64 czerwonych diod, które połączone są w taki sposób, że tworzą ze sobą kolumny oraz rzędy. Jest to połączenie w konfiguracji wspólna katoda. Takie rozwiązanie redukuje liczbę pinów potrzebnych do kontroli wyświetlacza, ale wymaga zastosowania sterowania dynamicznego. Maksymalny możliwy prąd jaki może przepływać przez jeden segment matrycy wynosi 25mA. Ale w przypadku sterowania dynamicznego przy współczynniku wypełnienia $\frac{1}{10}$ i częstotliwości 1KHz zwiększa się do wartości 100mA. Nominalnie napięcia przewodzenia przy 20mA pojedynczego segmentu wynosi 2V.

Jako element zapewniający komunikację między matrycami składającymi się na wyświetlacz, a mikrokontrolerem wybrany został układ firmy Maxim Integrated o oznaczeniu MAX7219. Układ ten jest sterownikiem wyświetlaczy o wspólnej katodzie i wykorzystuje interfejs SPI do komunikacji. Sterownik może być użyty do sterowania siedmiosegmentowymi wyświetlaczami, zawierającymi aż osiem niezależnych cyfr. W strukturze układu zawarty jest dekodery kodu BDC,

64 bajty pamięć statycznej RAM w której przechowywane są stany poszczególnych diod. Tylko jeden zewnętrzny rezystor wymagany jest by ustalić prąd dla wszystkich diod LED w matrycy. Wygodny trzyprzewodowy interfejs łączy się ze wszystkimi powszechnymi mikroprocesorami. Konstrukcja sterownika pozwala na uaktualnianie pojedynczej cyfry bez ponownego przepisania pozostałych cyfr wyświetlacza. Układ MAX7219 także pozwala użytkownikowi na wybranie trybu pracy z dekodowaniem kodu BCD lub nie dla każdej cyfry. Układ posiada niskoprądowy tryb uśpienia, analogową i cyfrową kontrolę jasności, a także tryb testowy, który włącza wszystkie LED na wyświetlaczu.

Do przetestowania pracy matrycy i napisania funkcji obsługujących wyświetlacz wykorzystana została płytko uruchomieniowa przedstawiona na rysunku Rys. 2. Płytko oparta jest mikrokontrolerze firmy Microchip PIC32MX470F512H, w obudowie QFN64. Posiada on 512 KB pamięci Flash przeznaczonej dla danych oraz 128 kb pamięci SRAM dla programu. Procesor może być taktowany z maksymalną częstotliwością 120MHz. Ponadto w swojej budowie zawiera zestaw timerów, obsługę przerwań, możliwość komunikacji poprzez interfejsy SPI, I2C, UART, zegar czasu rzeczywistego, 10 bitowy przetwornik analogowo cyfrowy, komparator sygnałów.



Rys. 2. płytko uruchomieniowa

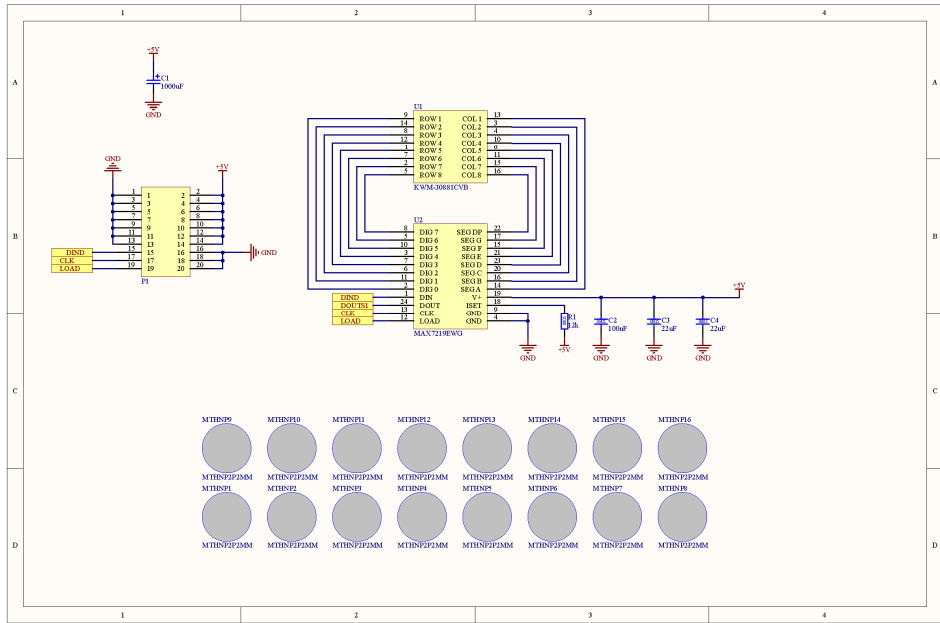
Do komunikacji między mikrokontrolerem, a układem MAX7219 wykorzystywany jest protokół SPI. Aby rozpocząć komunikację najpierw na linii wyboru musi pojawić się stan niski, a następnie w cykl zegara taktującego wysłane jest 16

bitów w których zawarty jest adres rejestru sterownika oraz wartość jaką chcemy tam wpisać. Na sam koniec stan na linii wyboru wraca do wartości wysokiej i komunikacja zakańcza się. Poniżej przedstawiona została mapa rejestrów sterownika oraz format jednej ramki danych:

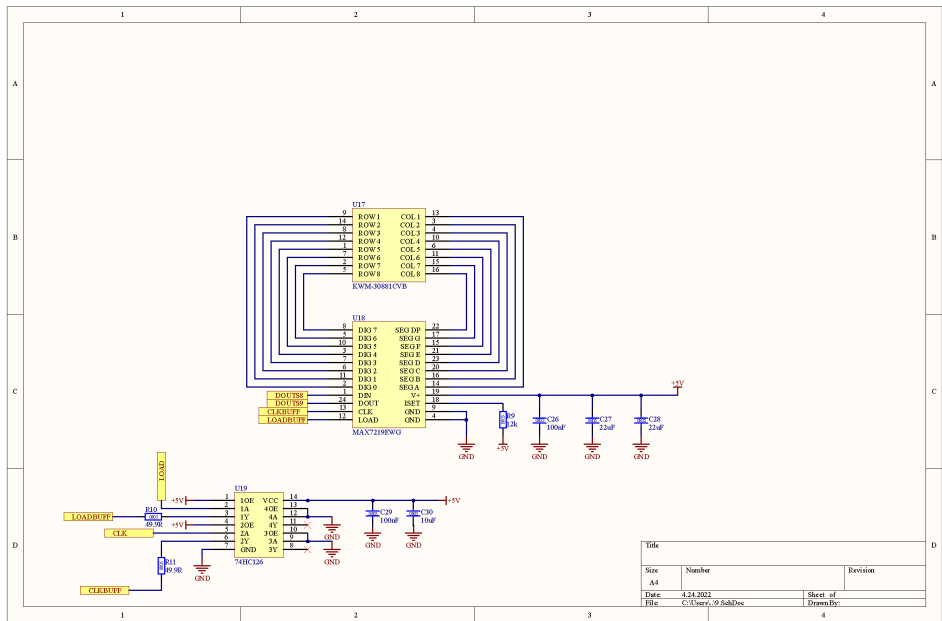
REGISTER	ADDRESS					HEX CODE
	D15-D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Dsisplay Test	X	1	1	1	1	0xXF

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	x	ADDRESS				MSB DATA				LSB			

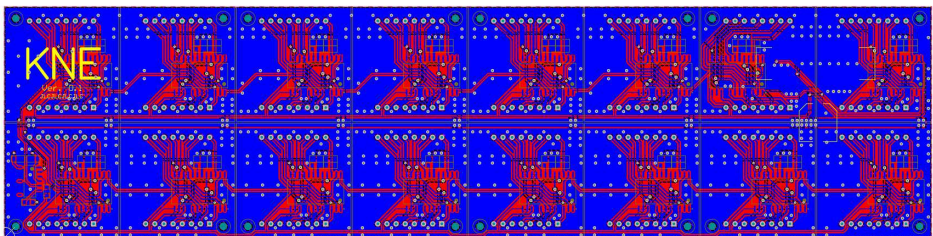
Finalny efekt decyzji podjętych podczas wstępnej fazy projektowania wyświetlacza widzimy na grafice poniżej w formie gotowego schematu ideowego wykonanego w programie Altium Designer 21. Wyświetlacz składa się z 16 identycznych sekcji pokazanych na poniższych rysunkach. W celu uproszczenia wyróżnione są tylko trzy sekcje, ponieważ pozostałe 13 sekcji jest identycznych jak na Rys.2. Zasilanie oraz sygnały sterujące doprowadzone są do płytki poprzez 24 pinowe złącze IDC. Na całej długości płytki wyświetlacza znajdują się otwory montażowe o średnicy 2,2MM. Do każdego układu MAX7219 dołączone są kondensatory odsprężające o wartościach ukazanych na poszczególnych rysunkach. Układ 74HCT126 pełni rolę bufora poprawiającego integralność sygnału.



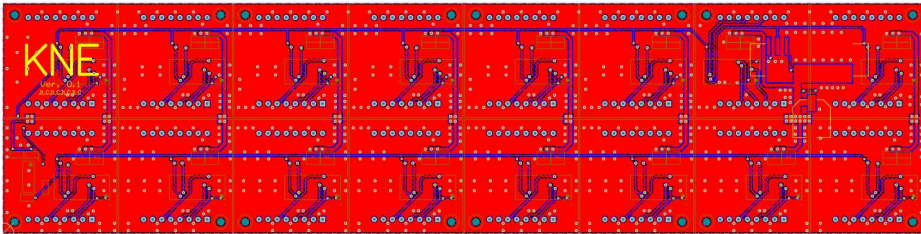
Rys. 3. Schemat ideowy na którym zawarte są: otwory montażowe, złącze IDC, pojedyncza sekcja z układem MAX7219 i matrycą LED.



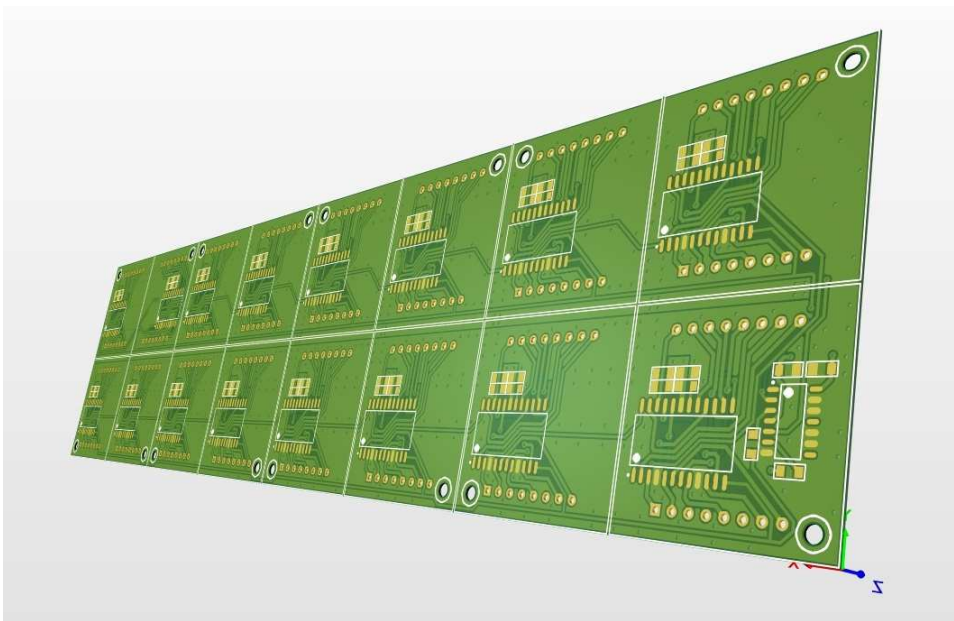
Rys. 5. Schemat ideowy na którym znajduje się pojedyncza sekcja z układem MAX7219 i matrycą LED oraz układ sterownika linii 74HC126



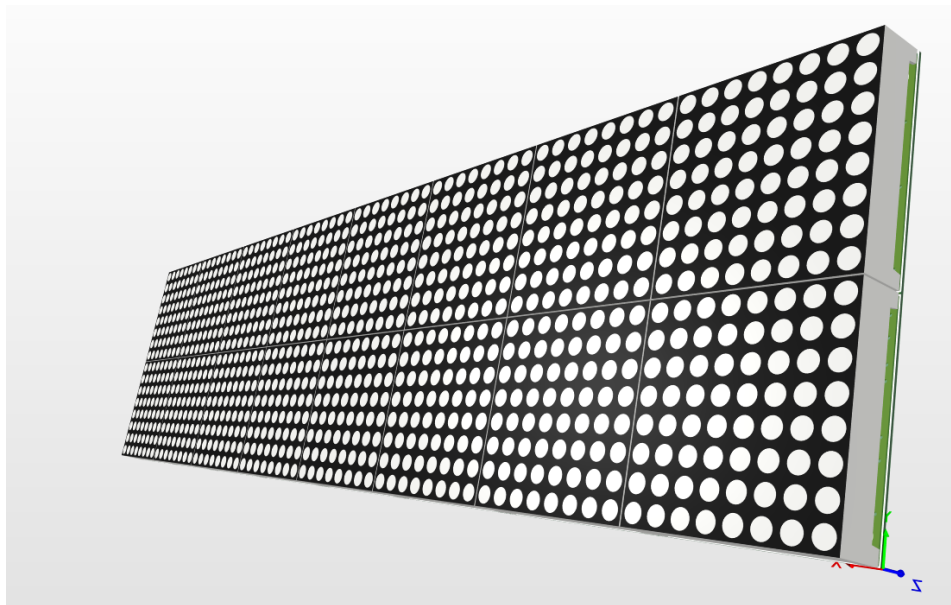
Rys. 6. Projekt płytki PCB – widok na dolną warstwę



Rys. 7. Projekt płytki PCB – widok na górną warstwę



Rys. 8. Projekt płytki PCB – widok w perspektywie 3D bez elementów



Rys. 9. Projekt płytki PCB – widok w perspektywie 3D z elementami

Program testowy napisany został na mikrokontroler PIC32MX470F512H przy pomocy środowiska programistycznego MPLABX. Aby funkcje działały na innym mikrokontrolerze wymagają przeportowania. Kod źródłowy zawarty jest w trzech plikach: font.c, main.c, functions.c. W pierwszym pliku znajduje się czcionka wygenerowana za pomocą strony internetowej „Led Matrix Editor”. Czcionka przechowywana jest w dwu wymiarowej tablicy o nazwie Font. Zawiera się w niej cały alfabet ASCII. Poniżej przedstawiona została część pliku font.c:

```
#include<xc.h>
const uint8_t Font[][8] = {
{
0b00011000,
0b00111100,
0b00111100,
0b00011000,
0b00011000,
0b00000000,
0b00011000,
0b00000000
},
{
0b01010000,
0b01010000,
```

```

0b01010000,
0b00000000,
0b00000000,
0b00000000,
0b00000000,
0b00000000
},
...

```

W pliku `functions.c` znajdują się funkcje obsługujące wyświetlacz oraz konfigurujące mikrokontroler. Lista funkcji:

- `void SPI_config()` – konfiguracja interfejsu SPI w trybie MODE 0,
- `void Display_config()` – konfiguracja sterownika MAX7219,
- `void display_clear()` – wyczyszczenie wyświetlacza,
- `void Print_at_X_Y(uint8_t x, uint8_t y, uint8_t symbol)` – funkcja wyświetla znak na wyświetlaczu o zadanych koordynatach. Znak jest podawany w standardowym zapisie języka C czyli w pojedynczym cudzysłowie. Na przykład, wywołanie funkcji z argumentami `x = 10, y = 10, znak = 'A'` wyświetli na wyświetlaczu literę „A” na koordynatach `x = 10, y = 10`,
- `void Send_data_to_display(uint8_t byte_1, uint8_t byte_2)` – wysłanie danych do wyświetlacza
- `void SPI1_write_to_buffer(uint8_t byte)` – przesłanie danych do bufora SPI, funkcja używana jest przez funkcje `SPI1_send_data()`,
- `void SPI1_send_data(uint8_t byte_1, uint8_t byte_2)` – wysłanie 16 bitów do sterownika MAX7219,
- `uint8_t Flip_byte(uint8_t data)` i `uint8_t Conversion(uint8_t data)` – funkcje pomocnicze.

Przykładowe użycie funkcji na mikrokontrolerze PIC32MX470F512H:

```

...
void main(void) {
    //SYSCLOCK = 40MHz
    //PBCLK = SYSCLOCK / 2

    ANSELA = 0; //Odłączenie wszystkich portów od przetwornika
    ADC
    ANSELB = 0;

    RPB13Rbits.RPB13R = 0b0011; //RPB13 jako SDO1
    TRISBbits.TRISB13 = 0; //RB13 jak wyjście SDO1
    TRISBbits.TRISB14 = 0; //RB14 jako wyjście SCK1
    TRISBbits.TRISB15 = 0; //RB15 jak wyjście OE1

```

```
//konfiguracja T32
T2CON = 0;
T3CON = 0;
T2CONbits.TCKPS = 0b0111; //Częstotliwość licznika wynosi
PBCLK / 256
T2CONbits.T32 = 0b0001; //Liczniki T3 oraz T2 pracują ra-
zem jako licznik 32 bitowy
PR2 = 0xFFFF; //Przepełnienie następuje przy wartości
0xFFFFFFFFFFFF
PR3 = 0xFFFF;
T2CONbits.ON = 1; //Licznik zostaje uruchomiony

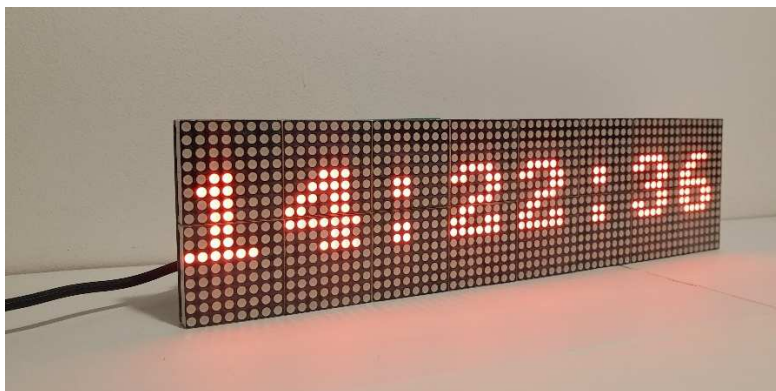
//Poczekaj aż wartość licznika osiągnie liczbę 78125. Pro-
gram czeka w tym miejscu przez 1000ms
while ((TMR3 << 16 | TMR2) <= 78125) {
}

T2CONbits.ON = 0; //Licznik zostaje wyłączony
TMR2 = 0; //Wyzerowanie wartości licznika
TMR3 = 0;

SPI_config(); //Konfiguracja interfejsu SPI1
Display_config(); //Konfiguracja sterowników MAX7219
Display_clear(); //Wyczyszczenie wyświetlacza

//Wyświetlenie przykładowego tekstu na wyświetlaczu.
Print_at_X_Y(0, 4, '1');
Print_at_X_Y(8, 4, '4');
Print_at_X_Y(16, 4, ':');
Print_at_X_Y(24, 4, '2');
Print_at_X_Y(32, 4, '2');
Print_at_X_Y(40, 4, ':');
Print_at_X_Y(48, 4, '3');
Print_at_X_Y(56, 4, '6');
//Na wyświetlaczy pojawi się napis "14:22:36"
Send_data_to_display(); //Wysłanie danych do wyświetlacza

while (1) {
}
}
...
```



Rys. 10. Rzeczywisty wyświetlacz wyświetlający przykładową godzinę

Kod źródłowy umieszczony został na platformie GitHub i nie jest w większości opisany. Link do platformy GitHub:

https://github.com/HaaX21/matrix_display(dostęp:29.04.2022).

2. MOŻLIWOŚĆ ROZWINIĘCIA PROJEKTU

Opisywany w artykule wyświetlacz przemysłowy, dzięki prostocie swojej konstrukcji i swoim dużym możliwościom co do wyświetlania znaków, może zostać w prosty sposób wykorzystany przy innych projektach wymagających do swojego działania wyświetlania komunikatów dla użytkownika.

W planach rozwojowych naszego projektu będzie rozwinięcie jego funkcji, takich jak wyświetlanie aktualnej godziny, dzięki zastosowaniu wbudowanego w mikrokontroler PIC32MX470 zegara czasu rzeczywistego. W dalszych planach jest także stworzenie obudowy dla wyświetlacza, zostanie ona wydrukowana na drukarce 3D. Ponadto oprócz wyświetlania aktualnego czasu chcemy dodać także wyświetlanie parametrów środowiskowych takich jak aktualna temperatura, wilgotność powietrza, ciśnienie atmosferyczne i tym podobnych. Osiągnąć to można poprzez dołączenie odpowiednich czujników do mikrokontrolera zarządzającego pracą wyświetlacza, oraz napisaniu programu do ich obsługi. Innym sposobem na zwiększenie funkcjonalności wyświetlacza byłoby sterowanie nim za pomocą komputera klasy PC w celu przesłania wyników z różnych urządzeń pomiarowych podłączonych do komputera na wyświetlacz.

3. PODSUMOWANIE

Wyświetlacz przemysłowy oparty na matrycach LED, jest urządzeniem, którego najważniejszą cechą jest prosta zwarta konstrukcja oraz duże możliwości wyświetlania znaków. Dzięki swojej prostocie nie wymaga on wielu elementów do działania i jest względnie tani a także nie pobiera on przy standardowej pracy wiele mocy. Może z powodzeniem być stosowany przy innych projektach jako element zapewniający komunikacje urządzenia z użytkownikiem.

ŹRÓDŁA INTERNETOWE

1. <https://pdf1.alldatasheet.com/datasheet-pdf/view/73745/MAXIM/MAX7219.html>(dostęp: 29.04.2022).
2. 29.04.2022).
3. <https://www.alldatasheet.com/datasheet-pdf/pdf/515133/MICRO-CHIP/PIC32MX470.html>(dostęp:29.04.2022).
4. <https://inthou.pl/protokoly-komunikacyjne-spi-teoria/>(dostęp:29.04.2022).
5. <https://www.luckylight.cn/media/component/data-sheet/KWM-30881CVB.pdf>(dostęp: 29.04.2022).
6. 29.04.2022).
7. <https://xantorohara.github.io/led-matrix-editor>(dostęp:29.04.2022).

Łukasz SZKARADEK, Krzysztof SADOWY, Patryk JASKUŁA, Hubert SKALSKI

dr inż. Mirosław SOBASZEK
opiekun naukowy

OBSŁUGA TENSOMETRU ZA POMOCĄ MIKROKONTROLERA STM32 NA PRZYKŁADZIE WAGI STOŁOWEJ

Artykuł opisuje sposób odczytywania danych z czujnika tensometrycznego przy pomocy mikrokontrolera STM32. Sygnał analogowy z czujnika trafia na wzmacniacz operacyjny HX711, z którego dane w postaci cyfrowej trafiają do mikrokontrolera, gdzie odpowiednio przetworzone są wyświetlane na wyświetlaczu LCD. Cały układ został umieszczony w specjalnie zaprojektowanej obudowie stworzonej na wzór wagi stołowej, wykonanej przy użyciu techniki druku 3D.

Słowa kluczowe: waga, stm32, tensometr, LCD

WPROWADZENIE

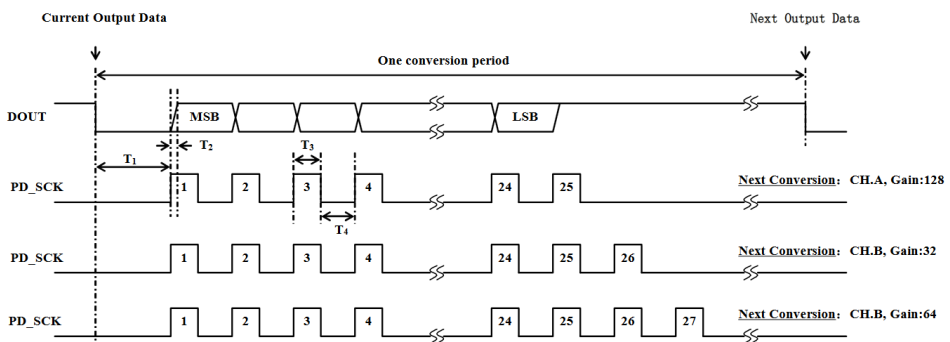
Tensometr jest miernikiem służącym do pomiaru naprężenia jakie występuje na powierzchni badanego przedmiotu, do którego ów tensometr jest przyklejony. Wykorzystana w projekcie belka tensometryczna jest przykładem tensometru elektrooporowego, oznacza to, że odkształcenie względne belki jest wprost proporcjonalne do względnego przyrostu rezystancji tensometru. Najważniejszymi parametrami tensometru jest współczynnik proporcjonalności k który wynika z materiału jakiego został stworzony tensometr oraz maksymalna wartość siły jaką możemy przyłożyć do czujnika. Współczesne zastosowanie tensometrów jest bardzo szerokie, tensometry wykorzystuje się do pomiaru naprężeń występujących w różnych konstrukcjach budowlanych, czy przy produkcji przetworników wagi, siły i momentu obrotowego.

Celem projektu było stworzenie gotowego urządzenia, ukazującego możliwości tensometrów oraz zaimplementowanie programu pozwalającego na dokładny odczyt oraz prezentację danych.

1. ELEMENTY WYKORZYSTANE W PROJEKCIE

Za cały układ odpowiada mikrokontroler STM32F103C8T6 Bluepill o taktowaniu 72MHz, który posiada 37 portów wejścia/wyjścia oraz jest zasilany napięciem +5V. Komunikacja mikrokontrolera z wyświetlaczem będzie się odbywała za pomocą magistrali I2C, natomiast komunikacja ze wzmacniaczem HX711 będzie się odbywać za pomocą specjalnego formatu danych zdeklarowanych przez producenta układu.

Belka tensometryczna użyta w projekcie posiada zakres pomiarowy wynoszący 2kg, zasilanie oraz odczyt napięcia jest realizowany po przez wzmacniacz operacyjny HX711. Układ ten zasilany jest napięciem 2,6-5V i posiada wzmacniacz różnicowy który pozwala odczytywać wartości napięcia na wyjściu belki tensometrycznej. HX711 posiada również 24 bitowy przetwornik analogowo-cyfrowy dzięki któremu przetworzone dane w postaci cyfrowej trafiają do mikrokontrolera. Do komunikacji mikrokontrolera z modułem służą jedynie dwa wyprowadzenia w postaci linii danych oraz linii zegarowej. Sposób w jaki odbywa się komunikacja oraz odpowiednie czasy zostały przedstawiony na ilustracji 1.



Rysunek 1. Sposób komunikacji ze wzmacniaczem HX711.

Symbol	Note	MIN	TYP	MAX	Unit
T ₁	DOUT falling edge to PD_SCK rising edge	0.1			μs
T ₂	PD_SCK rising edge to DOUT data ready			0.1	μs
T ₃	PD_SCK high time	0.2	1	50	μs
T ₄	PD_SCK low time	0.2	1		μs

Rysunek 2. Sposób komunikacji ze wzmacniaczem HX711, czasy odpowiedzi

Użyty wyświetlacz LCD posiada dwie linie po 16 znaków w linii każda, posiada on także wewnętrzny sterownik HD44780, oparty na układzie PCF8574. Sam wyświetlacz posiada 16 pinów, które odpowiadają za jego zasilanie oraz komunikację w celu wyświetlenia zadanych znaków. Korzystanie z wyświetlacza w bezpośredni sposób byłoby czasochłonne i znacznie utrudnione, zważywszy na liczbę przewodów które należałoby podłączyć pomiędzy ekranem a przedstawionym powyżej mikrokontrolerem. W związku z tym zastosowany został konwerter I2C oparty o układ PCF8574 podłączany bezpośrednio pod piny wyświetlacza. To „ulepszenie” powoduje redukcję liczby prowadzonych połączeń z 16 na 4. Wśród zredukowanej liczbie przewodów znajdziemy wyprowadzenia na zasilanie +5V i GND oraz dwa, (SDA-linia danych, SCL-linia zegarowa) na komunikację przy pomocy magistrali szeregowej I2C.

2. ŚRODOWISKO PROGRAMISTYCZNE I UŻYTE BIBLIOTEKI

Program sterujący został napisany w dedykowanym przez producenta środowisku programistycznym STM32 Cube IDE w języku C. W projekcie została użyta wysokopoziomowa biblioteka HAL zapewniająca API (Application Programming Interfaces) do łączności z układami peryferyjnymi mikrokontrolera. Dodatkowo została stworzona własna biblioteka do obsługi wyświetlacza LCD.

3. IMPLEMENTACJA PROGRAMU

Po włączeniu wagi, na której nie został umieszczony żaden przedmiot na górnym segmencie wyświetlacza pojawi się napis „WAGA” natomiast dolny segment będzie stale wyświetlał wartość 0 do momentu, w którym czujnik wykryje nacisk. Wówczas odczytany wynik zostaje wyświetlony na dolnym segmencie wyświetlacza i jest wyświetlany do momentu zmiany siły nacisku (napisane wyniku) lub jej całkowitego braku(wyczyszczenie wyświetlacza).

Funkcja `weight_reading` jest odpowiedzialna za odczytanie wyniku pomiaru przy użyciu przetwornika analogowo-cyfrowego HX711. Po ustabilizowaniu linii danych oraz ustawieniu poziomu niskiego dla linii zegara, wartość zmiennej jest zerowana a następnie w pętli `while` następuje moment oczekiwania na możliwość przyjęcia danych. Jeżeli na linii danych wystąpi stan niski następuje odczytanie 24 bitów danych. Po 25 impulsie zegarowym wynik jest konwertowany po przez różnicę symetryczną(XOR) po czym następuje zwrócenie gotowego wyniku.

```

uint32_t weight_reading(void){
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
/*Ustabilizowanie linii danych*/
delay_us(1);
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET); /*Ustawienie
poziomu niskiego na lini zegara*/
delay_us(1);
weight = 0.0; /*Wyzerowanie zmiennej*/
while(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_6) == GPIO_PIN_SET){}
/*Czekanie na możliwość pobrania danych*/
for(i=0;i<24;i++){
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET); /*Ustawienie
poziomu wysokiego na lini zegara*/
delay_us(1);
weight = weight << 1; /*Przesunięcie bitowe o jeden w lewo*/
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET); /*Ustawienie
poziomu niskiego na lini zegara*/
delay_us(1);

if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_6) == GPIO_PIN_SET){
/*Odczytanie jednego bitu danych*/
weight ++; /*Inkrementacja zmiennej*/
}
}
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET); /*Ustawienie
poziomu wysokiego na lini zegara*/
delay_us(1);
weight=weight ^ 0x800000; /*XOR na masce 800000h*/
HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
/*Ustawienie poziomu niskiego na lini zegara*/
delay_us(1);
return weight; /*Zwracanie wyniku*/
}

```

Listing 1. Deklaracja funkcji `weight_reading` służąca do odczytu wartości.

W celu uzyskania precyzyjnego odmierzenia czasu pomiędzy kolejnymi krokami w funkcji `weight_reading`, został użyty timer. Poniżej znajduje się funkcja, która pozwala na tworzenie opóźnień w mikrosekundach.

```

void delay_us (uint16_t us){
__HAL_TIM_SET_COUNTER(&htim2,0); /*Zerowanie
wartości licznika*/
while (__HAL_TIM_GET_COUNTER(&htim2) < us);
/*Oczekiwanie aż licznik zliczy do danej wartości*/
}

```

Listing 2. Deklaracja funkcji timera służąca do tworzenia opóźnień.

Główna pętla programu zawiera funkcję uruchamiającą timer oraz trzy instrukcje warunkowe. Pierwsza z nich jest wykonywana, jeżeli ważony przedmiot jest lżejszy niż 1kg, wówczas następuje wypisanie wyniku w gramach na wyświetlacz oraz przejście do etykiety ODCZYT, w której uzyskany wynik jest porównywany z aktualnym stanem – w przypadku zmiany obciążenia wyniki zostanie uaktualniony, w przeciwnym razie odczytana wartość będzie w sposób ciągły widniała na wyświetlaczu. Analogicznie program postąpi w przypadku drugiej instrukcji warunkowej, jedyną zmianą jest przeliczenie jednostki na kilogramy z dokładnością do 3 miejsc po przecinku. Trzecia instrukcja warunkowa jest odpowiedzialna za stan, w którym na wadze nie jest umieszczony żaden przedmiot. Dodatkowo w drugiej i trzeciej instrukcji występuje zabezpieczenie przed „fałszywymi” danymi, spowodowanymi choćby odgięciem belki tensometrycznej w przeciwną stronę.

```
while (1){
HAL_TIM_Base_Start(&htim2); /*Uruchomienie timera*/
printf("waga wynosi 0 \n");
HAL_Delay(1000);
ODCZYT:
    value=weight_reading(); /*Odczyt wagi*/
    if(value == weight_next){ /*Porównanie wartości*/
        HAL_Delay(1000);
        goto ODCZYT;
    }
if(value>0 & value<1000){ /*Pierwsza instrukcja warunkowa*/
/*Wyświetlenie wartości*/
    sprintf((char*)disp.f_line, "    WAGA:");
    sprintf((char*)disp.s_line, "    %lu g", value);
    lcd_display(&disp);
    HAL_Delay(1000);
    weight_next = value;
    goto ODCZYT;
}
if(value>1000 & value<10000){ /*Druga instrukcja warunkowa*/
    value_count=value/1000.0;
/*Wyświetlenie wartości*/
    sprintf((char*)disp.f_line, "    WAGA:");
    sprintf((char*)disp.s_line, "    %.3f Kg", value_count);
    lcd_display(&disp);
    HAL_Delay(1000);
    weight_next = value;
    goto ODCZYT;
}
if(value==0 & value<10000){ /*Trzecia instrukcja warunkowa*/
    lcd_clear(&disp); /*Wyczyszczenie wartości*/
    sprintf((char*)disp.s_line, "    0 g");
    lcd_display(&disp);
    HAL_Delay(1000);
    weight_next = value;
    goto ODCZYT;
}
```

```

    }
    HAL_Delay(1000);
    goto ODCZYT;
}

```

Listing 3. Deklaracja głównej pętli programu.

W projekcie wykorzystana została autorska biblioteka `lcd_i2c`, która pozwala na komunikację z wyświetlaczem. W momencie otrzymania wyniku z belki tensometrycznej, przetworzonego oraz przeliczonego na wartość rozumianą dla użytkownika, należy wyświetlić go na ekranie. To właśnie zapewnia napisana biblioteka, która odpowiednio inicjuje ekran w trybie 4 lub 8 bitowym, ponadto zapewnia odświeżanie a także czyszczenie przestrzeni którą zajmują wyświetlane znaki.

```

#ifndef INC_LCD_I2C_H_
#define INC_LCD_I2C_H_
#include <stdbool.h>
#include <stdint.h>

#define HI2C_DEF hi2c1 /*Zdefiniowanie używanego interfejsu*/
/*Zdefiniowanie używanych pinów*/
#define RS_PIN 0x01
#define RW_PIN 0x02
#define EN_PIN 0x04
#define BL_PIN 0x08

#define INIT_8_BIT_MODE 0x30
/*Inicjalizacja trybu 8-bitowego*/
#define INIT_4_BIT_MODE 0x20
/*Inicjalizacja trybu 4-bitowego*/

#define CLEAR_LCD 0x01
/*Komenda odpowiedzialna za wyczyszczenie zawartości
wyświetlacza*/
/*Zdefiniowanie sposobu graficznego przedstawienia kursora*/
#define UNDERLINE_OFF_BLINK_OFF 0x0C
#define UNDERLINE_OFF_BLINK_ON 0x0D
#define UNDERLINE_ON_BLINK_OFF 0x0E
#define UNDERLINE_ON_BLINK_ON 0x0F
/*Opis(zdefiniowanie) pierwszej oraz drugiej linii ekranu*/
#define FIRST_CHAR_LINE_1 0x80
#define FIRST_CHAR_LINE_2 0xC0

struct lcd_disp {
    uint8_t adres; /*Adres wykorzystywanego wyświetlacza*/
    char g_linia[17]; /*Oznaczenie pierwszej linii ekranu[ilość
znaków]*/
    char d_linia[17]; /*Oznaczenie drugiej linii ekranu [ilość
znaków]*/
}

```



```

        bool bl; /*Załączenie podświetlenia ekranu*/
};
/*Deklaracja użytych funkcji*/
void lcd_init(struct lcd_disp * lcd);
void lcd_write(uint8_t addr, uint8_t data, uint8_t xpin);

void lcd_display(struct lcd_disp * lcd);
void lcd_clear(struct lcd_disp * lcd);

#endif /* INC_LCD_I2C_H_ */

```

Listing 4. Deklaracja biblioteki lcd_i2c, plik c.

W kolejnym pliku biblioteki umieszczone zostały instrukcje ukazujące sposób wyświetlania całych wyrazów krok po kroku. Dodatkowo można w prosty sposób zaobserwować, jak wyświetlacz rozpoczyna pracę, a także kolejne kroki które zapewniają kolejno pobranie danych oraz wpisanie ich na ekran, czy sterowanie funkcją podświetlenia ekranu i jego czyszczeniem.

```

#include "lcd_i2c.h"
#include "stm32f1xx_hal.h"
#include "i2c.h"

void lcd_init(struct lcd_disp * lcd)
/*Przyjęcie struktury(wskaźnika) opisanej w plikulcd_i2c.h*/
{
    uint8_t xpin = 0;
    /* set backlight */
    if(lcd->bl) /*Podświetlenie ekranu wybór opcji*/
    {
        xpin = BL_PIN;
    }
    /*Aktywacja wyjścia odpowiedzialnego za podświetlenie ekranu*/
    /*Przy każdej komendzie ustawiamy podświetlenie na '1' aby utrzymać ciągłość podświetlenia ekranu*/
    HAL_Delay(40);
    lcd_write(lcd->adres, INIT_8_BIT_MODE, xpin);
    /*Trzykrotne wysłanie 'INIT_8_BIT_MODE'(0011)*/
    HAL_Delay(5);
    lcd_write(lcd->adres, INIT_8_BIT_MODE, xpin);
    HAL_Delay(1);
    lcd_write(lcd->adres, INIT_8_BIT_MODE, xpin);
    lcd_write(lcd->adres, INIT_4_BIT_MODE, xpin);
    /*Ustawienie trybu 4-bitowego*/

    lcd_write(lcd->adres, UNDERLINE_OFF_BLINK_OFF, xpin);
    /*Ustawienie braku podkreślenia kursora oraz braku jego mrugania*/

    lcd_clear(lcd); /*Czyszczenie ekranu*/
}
void lcd_write(uint8_t adres, uint8_t data, uint8_t xpin)

```

```

/*(adres; dane; dodatkowe flagi przełączające)*/
{
    uint8_t tx_data[4]; /*Utworzenie tablicy 4 elementowej w celu
    podzielenia danych na dwie części*/

    tx_data[0] = (data & 0xF0) | EN_PIN | xpin;
    /*Przesłanie danych w kolejności: od bitów najbardziej znaczących
    do najmniej znaczących(w tej linijce użycie adresu 0xF0 pomaga
    wyłuskać bity bardziej znaczące (ustawia 0 na mniej znaczących
    bitach)dane zostały zatrzaśnięte)*/
    tx_data[1] = (data & 0xF0) | xpin;
    /*Stan magistrali się nie zmienia, natomiast nie występuje już pin
    enable, dzięki czemu będzie stan niski co powoduje że nasz wyświet-
    lacz przyjmuje dane przesłane wcześniej*/

    tx_data[2] = (data << 4) | EN_PIN | xpin;
    /*Przesyłanie mniej znaczących bitów. Występuje przesunięcie bi-
    towe o 4 miejsca w lewo, zatrzaskujemy pin Enable*/
    tx_data[3] = (data << 4) | xpin; /*Zdjęcie pinu enable*/

    HAL_I2C_Master_Transmit(&HI2C_DEF, addr, tx_data, 4, 100);
    /*Złożenie naszego komunikatu w całość i przesłanie przy pomocy
    I2C (wskaźnik, adres, dane do przesłania{4 ponieważ będą 4 baj-
    ty},timeout=100ms)*/

    HAL_Delay(5);/*Czas oczekiwania od 2-5ms tak aby ekran przyjął
    dane*/
}

void lcd_display(struct lcd_disp * lcd)
{
    uint8_t xpin = 0, i = 0;
    if(lcd->bl)
    {
        xpin = BL_PIN;
    }
    lcd_clear(lcd);/*Czyszczenie wyświetlacza*/

    lcd_write(lcd->adres, FIRST_CHAR_LINE_1, xpin);
    /*Rozpoczęcie wpisywania danych na ekran od pierwszej linii*/
    while(lcd->g_linia[i]){ /*Sprawdzenie niezerowości znaku(gdy
    będzie 0 oznacza to koniec napisu)*/
        lcd_write(lcd->adres, lcd->d_linia[i], (xpin | RS_PIN));
        /*Po pobraniu pierwszego znaku wysyłamy go na adres ekspandera,
        jednocześnie ustawiając pin RS na 1, co sygnalizuje zapis danych
        na ekranie w kodzie ascii*/
        i++;/*Zwiększenie indeksu o jeden, pobranie kolejnego zna-
        ku itd. do momentu, aż skończą się znaki tablicy f_Line*/
    }
    i = 0;
    /*Po wyświetleniu wszystkich znaków zerujemy indeks*/
    lcd_write(lcd->adres, FIRST_CHAR_LINE_2, xpin);
    /*Podanie informacji o rozpoczęciu zapisywania danych na ekranie
    od pierwszego znaku drugiej linii*/
}

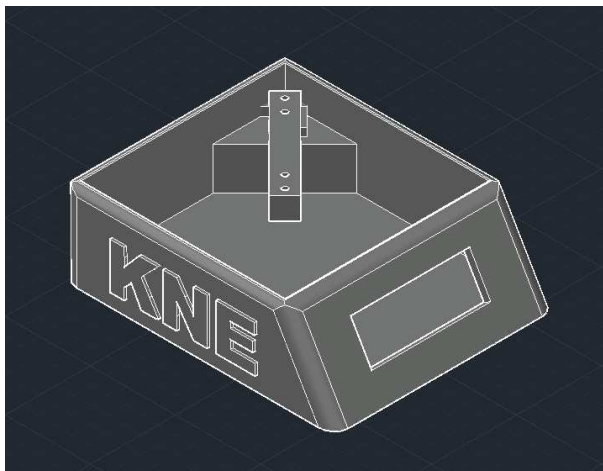
```

```
while(lcd->d_linia[i])
/*Analogicznie do poprzedniego przypadku sprawdzenie niezerowości
oraz zapis kolejnych znaków iteracyjnie */
{
    lcd_write(lcd->adres, lcd->d_linia[i], (xpin | RS_PIN));
    i++;
}
}
void lcd_clear(struct lcd_disp * lcd)
{
    uint8_t xpin = 0;
    if(lcd->bl) /*Włączenie podświetlenia*/
    {
        xpin = BL_PIN;
    }
    lcd_write(lcd->adres, CLEAR_LCD, xpin);
    /*Czyszczenie wyświetlacza*/
}
```

Listing 5. Deklaracja biblioteki lcd_i2c, plik h.

4. PROJEKT OBUDOWY

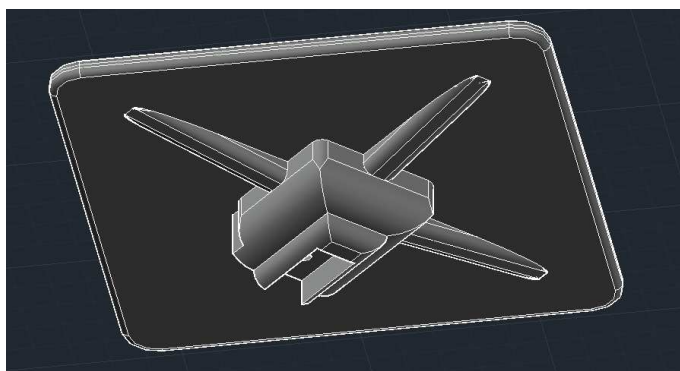
Obudowa urządzenia została zaprojektowana w programie AutoCad, składa się ona z dwóch części – głównej obudowy, w której są umieszczone wszystkie elementy oraz podstawki, na której będą umieszczane ważne przedmioty. Całość została wydrukowana przy użyciu drukarki 3D z PLA.



Rysunek 3. Zaprojektowana obudowa w programie AutoCAD.
Element główny, widok z przodu.



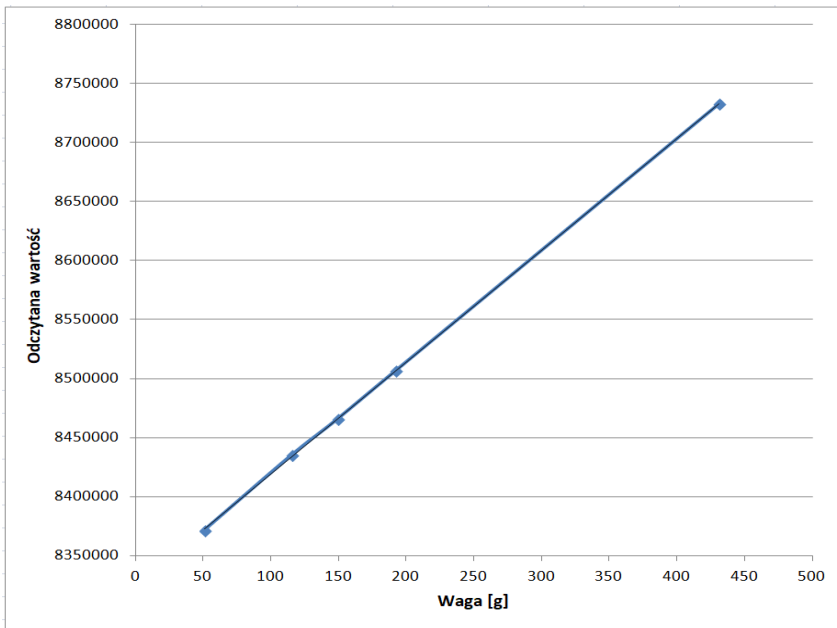
Rysunek 4. Zaprojektowana obudowa w programie AutoCAD. Element główny, widok z tyłu.



Rysunek 5. Zaprojektowana obudowa w programie AutoCAD. Element podstawy.

5. KALIBRACJA WAGI

Po uzyskaniu komunikacji z modułem HX711, otrzymane wyniki z przetwornika analogowo-cyfrowego należy przekształcić na odpowiednią jednostkę miary. W tym celu należy umieścić na belce tensometrycznej kilka przedmiotów o znanej nam wadze i na podstawie uzyskanych pomiarów sporządzić wykres.



Rysunek 6. Wykres zależności danych odbieranych przez mikrokontroler od wagi przedmiotów.

Stosunek wagi do odczytywanych wartości z przetwornika ma postać funkcji liniowej:

$$f(x) = ax + b$$

Dzięki temu możemy łatwo obliczyć współczynniki a,b które zostaną wykorzystane przy dekodowaniu mierzonej wartości:

$$f(x) = 943x + 8326243$$

W celu zwracania wyniku w określonej jednostce, do funkcji `weight_reading` został dopisany fragment kodu. Dzięki temu, wynik, który zwraca funkcja jest przedstawiony w gramach.

```
/*Działania arytmetyczne na wyniku*/  
wynik = weight - 8326243;  
wynik2 = (wynik /943.0) - 87; /*Uwzględnienie wagi podstawki*/  
  
return wynik2; /*Zwracanie wyniku*/
```

Listing 6. Kontynuacja kodu dla funkcji `weight_reading`.

6. PODSUMOWANIE

Założeniem projektu było stworzenie gotowego urządzenia pokazującego ideę możliwości zastosowania tensometru. Wykonany projekt można rozbudować i dostosować pod własne preferencje – przykładowo zmiana belki tensometrycznej i ponowna kalibracja wagi umożliwi powiększenie zakresu pomiarowego.



Rysunek 7. Prezentacja urządzenia.

ŹRÓDŁA INTERNETOWE

1. Dokumentacja modułu HX711, *24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales*, https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf (dostęp 28.04.2021) .
2. Informacje na temat tensometrów: <https://www.tensometry.com.pl/> (dostęp 28.04.2021) .
3. Dokumentacja ekranu LCD 2x16, *NHD-0216SZ-NSW-BBW-33V3*, <https://octopart.com/datasheet/nhd-0216sz-nsw-bbw-33v3-newhaven+display-21160410> (dostęp 28.04.2021).
4. Dokumentacja techniczna konwertera I2C, opartego na układzie PCF8574, https://botland.com.pl/index.php?controller=attachment&id_attachment=210 (dostęp 28.04.2021).



KOŁO

NAUKOWE

○ SOMETHING

ABOUT

IT SECURITY



Marek PIĘDEL

dr inż. Mariusz NYCZ
opiekun naukowy

JAK CEBULA WPŁYWA NA PRYWATNOŚĆ UŻYTKOWNIKA W INTERNECIE

Artykuł opisuje działanie sieci Tor potocznie nazywaną cebulową i jej wpływ na prywatność jej użytkowników. Tematyka funkcjonowania sieci tor jest obszarem, od którego wiele osób zaczyna interesować się prywatnością w sieci, czy ogólnie pojętym cyberbezpieczeństwem, w tym studenci koła naukowego Something about IT Security. Celem artykułu jest przybliżenie oraz opis historii rozwoju sieci tor, zaczynając od samej idei, dla której powstała, przechodząc do problemów, które należało rozwiązać, aby sieć wyglądała w ten sposób, w jaki wygląda obecnie. Co więcej, w artykule zawarte jest działanie głównego mechanizmu tej sieci, onion routing (trasowania cebulowego), korzystanie z przeglądarki Tor Browser, jak i kwestia finansowania organizacji The Tor Project, która odpowiada za obecny stan sieci. Ostatnim aspektem są sami użytkownicy, ich liczba oraz powody, dla których liczba osób zmienia się w różnych okresach wraz z danymi określającymi liczbę użytkowników w danych państwach.

Słowa kluczowe: onion routing, The Tor Project, Tor Browser.

WPROWADZENIE

Do tej pory niewiele technologii internetowych miało tak ogromny wpływ na anonimowy Internet, jak sieć, która została powszechnie nazwana jako sieć „Tor”. Sama przeglądarka „Tor Browser” jest zmodyfikowaną wersją popularnej przeglądarki internetowej Firefox. Korzystanie z tej przeglądarki Tor jest podobne do innych przeglądarek internetowych. To, co odróżnia to środowisko to fakt, że przeglądarka ta musi połączyć się z siecią Tor przed swoim uruchomieniem. Wszystkie te modyfikacje sprowadzają się do tego, aby osoby zewnętrzne, dostawcy internetowi i inne osoby, których zamiarem jest śledzenie połączenia lokalnego, nie mogły śledzić aktywności w Internecie osoby obranej za cel. Założeniem projektu sieci Tor zostało połączenie łatwości obsługi z efektywną anonimowością, z której praktycznie każdy może korzystać bez instrukcji technicznych. Sam geniusz przeglądarki Tor łączy w sobie łatwość obsługi bez wymogu działania oprogramowania do efektywnej pracy. Chociaż istnieją inne

sposoby anonimowego przeglądania Internetu, przeglądarka Tor jest zdecydowanie jedną z najprostszych w użyciu i jest do pobrania za darmo.

1. HISTORIA SIECI TOR

Projekt, jakim jest Tor Project, stał się organizacją non profit w 2006 roku, lecz sama idea „trasowania cebulowego” rozpoczęła się w latach 90. Sieć Tor stanowi grupa użytkowników, programistów, badaczy i założycieli, którzy zaangażowali się w jej tworzenie, choć stanowią zróżnicowaną grupę ludzi, mają wspólny cel i przekonanie, w którym każdy użytkownik Internetu powinien mieć prywatny dostęp do nieocenzurowanej sieci. Dynamiczny rozwój sieci w latach 90 wiązał się z koniecznością zabezpieczenia transmisji danych. Wraz z upływem czasu oczywistym stawały się kwestie dotyczące śledzenia i inwigilacji, choć same dane dotyczące danego użytkownika można było zaszyfrować, to pliki i same informacje, które przesyłał, zawierały w sobie metadane, czyli informacje pozwalające prześledzić ruch, jak i objętość danych związanych z ich wysłaniem.

W 1995 roku przedstawiciele Laboratorium Badań Marynarki Wojennej Stanów Zjednoczonych wyszli z rozwiązaniem mającym na celu stworzenie połączeń internetowych, które nie ujawniłyby, kto z kim rozmawia, nawet osobie monitorującej sieć. Ich pomysłem było stworzenie i wdrożenie pierwszych projektów badawczych i prototypów trasowania cebulowego. W efekcie w roku 1998 powstało gotowe teoretyczne opracowanie trasowania cebulowego. Projekt ten miał na celu zabezpieczenie komunikacji w otwartych sieciach między systemami militarnymi i wywiadowczymi, jak i zapewnienie anonimizacji ruchu operacyjnego. Na początku XXI wieku Roger Dingledine, świeżo upieczony absolwent Massachusetts Institute of Technology (MIT), rozpoczął wraz z Pauliem Syversonem pracę nad projektem Laboratorium Badań Marynarki Wojennej Stanów Zjednoczonych dotyczącym trasowania cebulowego. Nazwa projektu TOR, choć miała odróżnić ten oryginalny projekt od innych projektów dotyczących trasowania cebulowego 1, de facto stanowi akronim od The Onion Routing (pol. routing cebulowy)”.

Od momentu powstania w latach 90. ubiegłego wieku trasowanie cebulowe miało opierać się na zdecentralizowanej sieci. Z racji tego, że wymagała maksymalnego rozproszenia sieć ta, musiała być obsługiwana przez więcej niż jeden podmiot o różnych interesach i założeniach dotyczących zaufania, a oprogramowanie musiało być wolne i otwarte, aby zmaksymalizować przejrzystość i decentralizację. Dlatego też w październiku 2002 roku, kiedy sieć Tor została uruchomiona, jej kod został udostępniony na licencji wolnego i otwartego oprogramowania. Dostrzegając korzyści, jakie Tor przynosi prawom cyfrowym, Electronic Frontier Foundation (EFF) zaczęła w 2004 roku finansować prace Rogera i Nicka nad Torem. W kolejnych latach rozwiązanie to zaczęło zyskiwać popularność głównie wśród aktywistów i użytkowników zaawansowanych tech-

nicznie, ale nadal był trudny w użyciu dla osób mniej obeznanych technicznie, dlatego od 2005 roku zaczęto rozwijać narzędzia wykraczające poza samo proxy Tor. Wkrótce pojawiły się pakiety oprogramowania umożliwiające anonimowe korzystanie z sieci bez zaawansowanych umiejętności najbardziej popularnym wówczas, był pakiet łączący przeglądarkę Opera z programem Tor (OperaTOR), jednak ze względu na zamknięte źródła przeglądarki rosnące użycie technologii JavaScript nie oferował satysfakcjonującego poziomu bezpieczeństwa.

W 2006 roku założono Tor Project, jako organizację non-profit, której celem jest utrzymanie rozwoju sieci Tor. Wydania powstające w tym roku bazujące na przeglądarce Firefox umożliwiały blokowanie elementów javascriptowych, przez co pole do ataku na klienta zostało znacząco zmniejszone. Powstałe od 2006 roku kolejne wydania bazujące na przeglądarce Firefox znacząco zwiększyły bezpieczeństwo—dzięki blokadzie JavaScript powierzchnia ataku na klienta przez użycie złośliwego skryptu została znacząco zmniejszona. W 2008 roku rozpoczęto prace nad przeglądarką Tor Browser. Przeglądarka ta uczyniła Tor bardziej dostępnym dla zwykłych użytkowników Internetu i aktywistów.

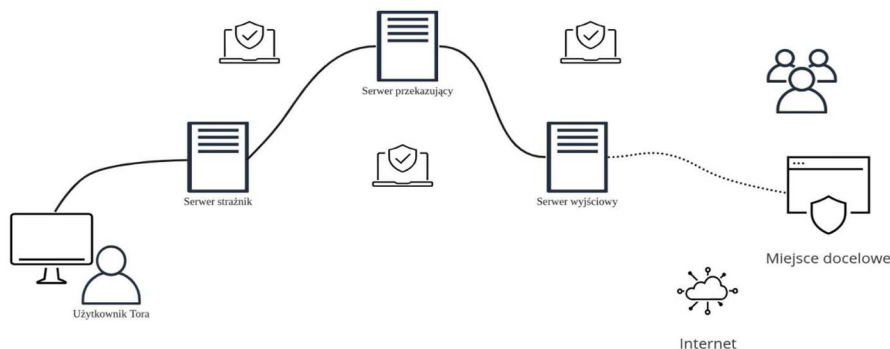
Od tej pory popularność sieci Tor ciągle rośnie, pośrednio wskutek rosnącej świadomości użytkowników Internetu, a w dużej mierze w wyniku zapotrzebowania na ochronę przed inwigilacją.

2. FUNKCJONOWANIE SIECI TOR

Działanie trasowania cebulowego zacznę od ogólnego przedstawienia, jak wygląda kierowanie ruchem w sieci Tor. Ruch kierowany jest przez wiele serwerów znajdujących się w różnych zakątkach świata. Każdy z nich, warstwy te można porównać do cebuli, stąd nazwa trasowania cebulowego. W efekcie czego nadawca pozostaje anonimowy, ponieważ poszczególne serwery widzą tylko, kto zdjął tę warstwę cebuli, czyli od kogo przyszedł dany pakiet i do kogo mają przesłać pakiet (cebule, której kolejna warstwa została obrana). Tor daje bardzo wysoki poziom anonimowości przy zachowaniu pewnych zasad.

Aby opisać, jak działa cały ruch, najpierw przejdźmy do tego, czym są poszczególne węzły, które są oznaczone na podanym rysunku. Pierwszym elementem są tak zwane serwery strażnicy, od nich rozpoczynamy całe połączenie, serwer ten odpowiada za odszyfrowanie pierwszej warstwy, w której widzi tylko informacje zawierające, gdzie ma przesłać kolejną część zaszyfrowanej informacji, którą mogą odczytać tylko kolejne serwery, jakimi są serwery przekazujące. Węzły te odbierają ruch od jednych z serwerów strażniczych i przekazują je dalej do serwerów przekazujących. Przekazywanie pomiędzy tymi serwerami może odbywać się wielokrotnie. Widzą one jedynie informacje na temat miejsca, w którym znajduje się trzeci węzeł. Trzecim węzłem są serwery wyjściowe, dopiero te serwery dysponują odpowiednimi kluczami, które umożliwiają obranie ostatniej warstwy cebuli, czyli sprawdzenie, gdzie tak naprawdę pakiet ma

trafić, służą one do nawiązania połączenia poza sieć Tor do części Internetu, która każdemu jest znana.



Rysunek 1. Przedstawienie działania trasowania cebulowego
 Źródło: <https://ourcodeworld.com/articles/read/953/how-to-route-all-the-machine-traffic-through-tor-in-kali-linux>

Oprócz tych serwerów istnieją serwery zwane katalogami, które zawierają listę innych serwerów i monitorują ich zachowanie, aby móc je odłączyć lub nie dopuścić do sieci w przypadku, gdy jakaś osoba lub firma zapragnęły kontroli nad serwerami, z którymi łączy się użytkownik. Osoby odpowiedzialne za Tor Project w takim przypadku kontrolują nagłe pojawienie się dużej ilości serwerów z jednego źródła. Nowe serwery są oceniane, jeżeli pojawia się ich zbyt duża ilość z jednego źródła, to nie są do sieci wpuszczane. Do dyspozycji są jeszcze serwery mostki, które są przydatne w sytuacji, gdy ktoś próbuje zablokować użytkownikowi dostęp do sieci Tor.

Przejdźmy teraz do omówienia całego działania krok po kroku. Na początku prosimy serwery katalogowe o listę wszystkich serwerów. Następnie algorytm po naszej stronie komputera wybiera sobie z nich serwer strażnik, serwery pośrednie oraz serwer wyjściowy. Całość ta działa na zasadzie obwodu z języka angielskiego „circle”. Po wybraniu serwerów zaczyna się projektowanie kawałek po kawałku naszej cebuli, czyli pakietu sieciowego szyfrującego każdą z warstw kluczem odpowiednim dla danego serwera. Zostaje wysłany pakiet do serwera strażnika, serwer ten obiera pierwszą warstwę cebuli, i widzi, że pozostała część, która dalej jest zaszyfrowana, ma wysłać do kolejnego serwera. Nie wie, co znajduje się w środku tej mniejszej części. Kolejny pakiet zostaje odebrany po stronie serwera przekaźnika, obiera on kolejną warstwę, do której ma specjalny klucz deszyfrujący, znowu odpakuje i przekazuje pakiet dalej. Takich operacji może być kilka. Oczywiście jest fakt, że im więcej tych operacji tym lepiej, lecz wiąże się to ze znacznym opóźnieniem połączenia. Jeśli jednak serwerów przekazujących jest ich za mało, ktoś może spróbować powiązać ruch

wejściowy z ruchem wyjściowym. Nie ma złotego środka. Jednak algorytmy wbudowane w większość oprogramowania korzystającego z sieci Tor zapewniają, że jest to losowane w pewnym dość rozsądnym zakresie. Serwer wyjściowy po odszyfrowaniu ostatniej warstwy przekazuje ją dalej.

Co z kwestią finansowania? Twórcy Tora od samego początku udostępnili swoje oprogramowanie na wolnej i otwartej licencji, a o jego rozwój dba fundacja non-profit. Od strony hardwarowej osobami wspierającymi są właśnie wolontariusze, zaś od strony software'owa lista sponsorów jest publikowana przez Tor Project, są tam zarówno instytucje rządowe z kilku różnych krajów głównie ze stanów zjednoczonych, są też firmy prywatne takie jak Google oraz fundacje takie jak Ford Foundation.

3. KORZYSTANIE Z SIECI TOR

Od początku powstania sieci Tor konfiguracja i użytkowanie sieci nie było prostym zadaniem do zrealizowania szczególnie dla osób mniej obytych technologicznie. Narzędziem, które umożliwiło dotarcie do większej ilości osób, stała się przeglądarka oparta na popularnej przeglądarce Firefox, wraz z pakietem specjalnych rozszerzeń, które mają za zadanie podnieść prywatność użytkownika. Przeglądarka ta jest cały czas rozwijana i jest jednym z najprostszych sposobów, by zacząć korzystanie z sieci Tor. Kolejnym rozwiązaniem jest Tails – czyli dystrybucja systemu GNU/Linux oparta na Debianie, która idzie o krok dalej od przeglądarki Tor Browser to kompletny system operacyjny uruchamiany z klucza USB. Dzięki temu może nie pozostawiać żadnych danych na dysku twardym. Jest skonfigurowany w sposób, który kładzie nacisk na prywatność. Nie zezwala również na połączenia inne, niż poprzez sieć Tor, czy to zainicjowane przez przypadek bądź wykorzystywane do ataku, którego celem jest odkrycie tożsamości użytkownika.

Użytkownik, który ma styczność z siecią Tor, na samym starcie natrafia na Hidden Services. Usługi te są usługami widocznymi tylko w sieci Tor i łatwo rozpoznać je po rozszerzeniu.onion. Ukryte usługi w założeniu zapewniają obustronną anonimowość. Serwer nie zna tożsamości klienta, a klient nie zna adresu ip serwera. Płyne z tego dodatkowa korzyść, brak znajomości adresu IP utrudnia wyłączenie usługi poprzez atak Denial of service, czyli atak, którego celem jest wyłączenie komputera lub sieci i uniemożliwienie dostępu do nich użytkownikom. Ataki DoS osiągają to, zalewając cel ruchem lub wysyłając mu informacje, które powodują awarię.

4. UŻYTKOWNICY SIECI Tor

Strona dotycząca projektu, jakim jest sieć Tor, udostępnia wiele statystyk dotyczących sieci. Jedną z takich właśnie statystyk jest liczba użytkowników w ostatnich kilku latach.



Rysunek 2. Statystyki dotyczące liczby użytkowników sieci Tor w ostatnich latach

Źródło: <https://metrics.torproject.org/userstats-relay-country.html>

Jak widać na załączonym wykresie liczba dwóch i pół miliona użytkowników, jest liczbą, wokół której jest formalna liczba użytkowników. Skoki w liczbie użytkowników wynikają z botnetów, czyli sieci składających się z dużej liczby komputerów, które zostały przejęte przez złośliwe oprogramowanie (malware), aby służyć osobie, która je stworzyła. Ruch ten wynika z chęci ukrycia połączeń pomiędzy komputerami w sieci Tor. Innymi powodami mogą być działania wojenne czy różne afery medialne związane z wyciekiem niepożądanych danych.

Na wykresie (rys. 3) można zaobserwować przykład wpływu działań wojennych na liczbę użytkowników sieci Tor. W kolejnych dniach poczynając od 24 marca różne media społecznościowe, stały się w Rosji częściowo zablokowane. Wzrosła więc liczba użytkowników w Rosji, czego wynikiem stał się wzrost liczby użytkowników, przekaźników i mostów na Ukrainie.



Rysunek 3. Statystyki dotyczące liczby użytkowników sieci Tor w ostatnich latach

Źródło: <https://metrics.torproject.org/userstats-relay-country.html>

Country	Mean daily users
United States	494787 (20.77 %)
Germany	219729 (9.23 %)
Russia	131367 (5.52 %)
India	103871 (4.36 %)
Finland	87846 (3.69 %)
Indonesia	83493 (3.51 %)
France	82515 (3.46 %)
Netherlands	70053 (2.94 %)
United Kingdom	67103 (2.82 %)
Brazil	51659 (2.17 %)

Rysunek 4. Statystyka przedstawiająca państwa z największą liczbą użytkowników

Źródło: <https://metrics.torproject.org/userstats-relay-table.html>

Jak widać największą liczbę użytkowników, posiadają Stany Zjednoczone, od których cała idea sieci Tor się zaczęła. Jak dokładnie przebiega proces zbierania takich danych dotyczących użytkowników w sposób szczegółowy, można znaleźć na stronie projektu.

W mediach najczęściej budowana jest negatywna narracja dotycząca użytkowników sieci Tor. W tym źródle informacji przedstawiani są jako przestępcy, którzy prowadzą bądź odwiedzają podziemne targowiska z substancjami nielegalnymi lub sprzętem militarnym. Jest to w większości prawda, jednak jest wiele

innych powodów, dla których ludzie z niego korzystają. Takim przykładami są dziennikarze, dla których jest to narzędzie, dzięki którym chronią tożsamości swoich źródeł i siebie samych. Następnym przykładem są obywatele mieszkający w państwach, w których dostęp do Internetu jest w różny sposób ograniczony, osoby te korzystają z sieci Tor, aby ominąć te ograniczenia. Sama sieć stała się ważnym narzędziem podczas Arabskiej wiosny, czyli protestów społecznych i konfliktów zbrojnych w krajach arabskich w latach 2010-2012. Nie tylko chroniła tożsamość ludzi w Internecie, ale także umożliwiała im dostęp do mediów społecznościowych i stron internetowych, które zostały zablokowane.

Organy wojskowe i organy ścigania również używają tej sieci do komunikacji i identyfikacji przestępców. W rzeczywistości, kiedy Tor był po raz pierwszy rozwijany w latach 90., głównym celem była ochrona komunikacji wywiadowczej. A zwykli użytkownicy, używają go, gdy chcą tylko poczuć, że robią wszystko, co w ich mocy, aby upewnić się, że nikt nie śledzi ich działań online.

Pozostaje teraz pytanie, w jaki sposób organy ścigania zatrzymują przestępców. W wielu przypadkach sprowadza się to do tego, że służby odpowiadające za bezpieczeństwo w danym kraju przejmują serwer ze stroną internetową i modyfikują ją w sposób pozwalający odkryć tożsamość odwiedzających, a udaje im się to przez na przykład brak hasła w koncie administratora serwera strony internetowej, co wydaje się oczywistą rzeczą, a jednak zdarza się często. Służby te w przeszłości wykorzystywały też atak polegającym na podstawieniu swoich węzłów sieci i atakowania tych, których nie kontrolowali, przez co ruch, leciał przez kontrolowane przez nich węzły.

5. PODATNOŚCI SIECI TOR

Podatności sieci Tor na zewnętrzne ataki nie tylko wykorzystywane są przez służby odpowiadające za bezpieczeństwo. Większość tych ataków przez analizę ruchu w sieci odróżnia wzorce przepływu danych od ruchu spowodowanego szyfrowaniem sieci Tor. Podobnie podczas tej analizy atakujący wykorzystują słabe punkty zaprojektowanych węzłów Tora i innych powiązanych usług, by następnie analizując poszczególne połączenia sieciowe potwierdzić swoje podejrzenia. Konieczne jest zatem podjęcie środków zaradczych w celu zmniejszenia ryzyka związanego z każdym atakiem. Środki zaradcze należy zastosować z perspektywy warstwy sieciowej, warstwy protokołu oraz warstwy aplikacji w celu ukrycia i usunięcia ruchu powiązanego z użytkownikiem w sieci. Wraz z upływem lat ataki stają się coraz bardziej skomplikowane, dlatego konieczna jest również weryfikacja certyfikatów i tworzenie tak zwanych kont przynęt, które pozwolą wykryć złośliwe węzły i umieścić je na czarnej liście węzłów wyjściowych sieci Tor. Wyścig w opracowywaniu nowych ataków, które zagrażają anonimowości użytkowników Internetu, będzie kontynuowany. Dlatego stworzenie solidnych podstaw teoretycznych jest kluczowym aspektem, aby można było dalej badać interakcje między atakami i środkami zaradczymi.

LITERATURA

1. Leonard Eddison, *Tor And The Deep Web: The Complete Guide To Stay Anonymous In The Dark Net* March 2018.
2. F. Shirazi, M. Goehring and C. Diaz, "Tor Experimentation Tools," *2015 IEEE Security and Privacy Workshops*, 2015, pp. 206-213, doi: 10.1109/SPW.2015.20.
3. A. Biryukov, I. Pustogarov and R. Weinmann, "Trawling for Tor Hidden Services: Detection, Measurement, De-anonymization," *2013 IEEE Symposium on Security and Privacy*, 2013, pp. 80-94, doi: 10.1109/SP.2013.15.
4. I. Karunanayake, N. Ahmed, R. Malaney, R. Islam and S. K. Jha, "De-Anonymisation Attacks on Tor: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2324-2350, Fourthquarter 2021, doi: 10.1109/COMST.2021.3093615.

ŹRÓDŁA INTERNETOWE

5. <https://www.torproject.org/>
6. <https://support.torproject.org/onionservices/>
7. <https://www.linuxjournal.com/content/tor-hidden-services>

○

KOŁO

NAUKOWE

○ „ESPRIT”

○

Klaudia STACHNIK

dr inż. Dorota GŁOWACZ–CZERWONKA, prof. PRZ
opiekun naukowy

UNIEPALNIONE, SZTYWNE PIANKI POLIURETANOWE NA BAZIE ROKOPOLU RF 151V. CZ. 1. WŁAŚCIWOŚCI FIZYCZNE

Sztywne pianki poliuretanowe cieszą się ogromnym zainteresowaniem ze względu na swoje wielowymiarowe zastosowanie w przemyśle. Cechują się one bardzo dobrymi właściwościami termoizolacyjnymi, ale również właściwościami fizycznymi. Celem tego artykułu jest zobrazowanie wpływu antypirenow addytywnych na te właśnie parametry. Skupiono się na badaniu gęstości pozornej, stabilności wymiarów, chłonności wody oraz współczynnika przewodzenia ciepła. Żadna z pianek nie wykazała znacznego pogorszenia się właściwości fizycznych, znalazły się też takie, których, wymienione wcześniej parametry, uległy polepszeniu. Wskazuje to na to, że otrzymane kompozycje piankowe mogłyby, z powodzeniem, być stosowane na skalę przemysłową.

Słowa kluczowe: sztywne pianki poliuretanowe, właściwości fizyczne, gęstość pozorna, stabilność wymiarów, chłonność wody, współczynnik przewodzenia ciepła.

WPROWADZENIE

Rok 1937 jest uznawany jako przełomowy w dziedzinie chemii polimerów. Wtedy właśnie niemiecki chemik, prof. Otto Bayer odkrył proces poliaddycji do syntezy poliuretanów z poliizocyjanianów i polioliu, który również za jego sprawą został opatentowany w tym samym roku. Dzięki temu udało się mu wraz ze swoimi współpracownikami w laboratorium w I.G w Leverkusen otrzymać materiał, który zrewolucjonizował przemysł chemiczny. Tym materiałem jest poliuretan. Mimo iż to wydarzenie miało miejsce wiele lat temu, odkrycie Bayera znacznie przyczyniło się do rozwoju tworzyw termoizolacyjnych¹.

Początkowo poliuretany były otrzymywane z diizocyjanianów i dioli. Na skalę przemysłową używano diizocyjanianu heksametylenu oraz glikolu butyle-

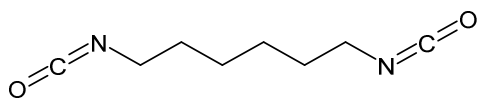
¹ Szycher M., *Szycher's handbook of polyurethanes*, CRC Press LLC, New York 1999, str. 1-2

nowego. Kolejny przełom w tematyce materiałów poliuretanowych nastąpił w 1950 roku, kiedy to zauważono ich potencjał i starano się udoskonalić ich strukturę. Z tego względu zaczęto stosować diizocyjania toluilenu (TDI) oraz poliiole poliestrowe i polieterole, co dało początek piankowym tworzywom poliuretanowym². Obecnie poliuretany są stosowane na szeroką skalę jako m.in. kleje, tworzywa piankowe, włókna poliuretanowe czy tworzywa skóropodobne.

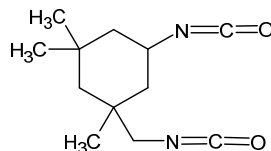
Poliuretany wyróżniają się na tle innych materiałów swoją prostotą w budowie strukturalnej. Początkowo składały się one z jednego lub dwóch monomerów, obecnie są to dużo bardziej rozbudowane struktury. Zawierają one trzy podstawowe monomery: diizocyjania, polioliol oraz związek odpowiadający za wzrost łańcucha^{1,3}. Istotnym reagentem, który odpowiada za powstanie struktury komórkowej, tak ważnej w procesie powstawania pianek poliuretanowych, jest porofor.

Izocyjania są jednymi z trzech związków, dzięki którym otrzymuje się poliuretany pod postacią pianek. W ich strukturze znajduje się charakterystyczna grupa $-N=C=O$. Ważnym aspektem jest to, aby użyty do reakcji izocyjania był wielofunkcyjny. Najlepszym wyborem są diizocyjania, które mogą być alifatyczne lub aromatyczne⁴.

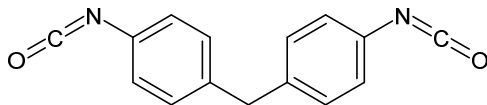
Do najczęściej stosowanych zalicza się: 1,6-diizocyjania heksametylenu (HDI), diizocyjania izoformonu (IPDI), 4,4'-diizocyjania difenylometanu (MDI), 2,4-diizocyjania toluilenu (2,4-TDI), diizocyjania dicykloheksylometylenu (HMDI)^{4,5}.



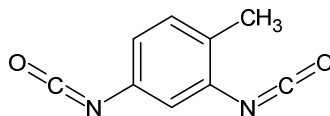
1,6-diizocyjania heksametylenu, **HDI**



Diizocyjania izoformonu **IPDI**



4,4'-diizocyjania difenylometanu **MDI**



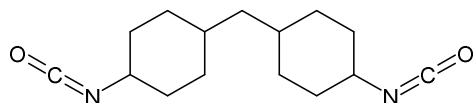
2,4-diizocyjania toluilenu, **TDI**

² Kricheldorf H.R., Nuyken O., Swift G., *Handbook of Polymer Synthesis second edition*. CRC Press, Nowy Jork, USA 2004, str. 503-505

³ Pivec T., Smole M. Sfiligoj, Gašparič P., Stana Kleinschek K., *Polyurethanes for Medical Use*, Tekstilec, 60 (4), 2017

⁴ Sardon H., Pascual A., Mecerreyes D., Taton D., Cramail H., Hedrick J. L., *Synthesis of Polyurethanes Using Organocatalysis: A Perspective*, *Macromolecules*, 48 (10), 2015, 3153-2165

⁵ Zwi A.B., *Isocyanates and health – a review*, *South African Medical Journal*, 67, 1985, str. 209-211



Diizocyjanian dicykloheksylometylenu,
HMDI

Chcąc wytworzyć strukturę sieciową, dzięki której produkt końcowy zyska na wytrzymałości, stosuje się multifunkcyjne izocyjaniany. Charakteryzują się one silnymi oddziaływaniami międzycząsteczkowymi, razem z przedłużaczami łańcucha są w stanie stworzyć łańcuch o strukturze sztywniejszej od polioli. W temperaturze pokojowej przejawiają właściwości typowo szkliste, z tego powodu nazywane są sztywnymi segmentami³.

Warto zaznaczyć fakt, że izocyjaniany są związkami toksycznymi, które szczególnie szkodliwy wpływ mają na układ oddechowy. Wykazują wysoką reaktywność, atom węgla występujący we wiązaniu $-N=C=O$ pełni rolę elektronu akceptorowego, azot jest silnym donorem, natomiast tlen nie posiada silnych właściwości donorowych³.

Kolejnymi niezbędnymi reagentami, dzięki którym możliwe jest otrzymanie poliuretanów, są poliole, inaczej alkohole wielowodorotlenowe. Mają one co najmniej dwie grupy hydroksylowe, co więcej, poliole mogą zawierać m.in. amidy, estry, akryle, etery, metale oraz inne grupy funkcyjne⁶.

Charakteryzują się one długimi, elastycznymi łańcuchami oraz ciężarem cząsteczkowym, mieszczącym się w granicach od 500 do 9000 jednostki masy atomowej. Wyróżnia się⁷

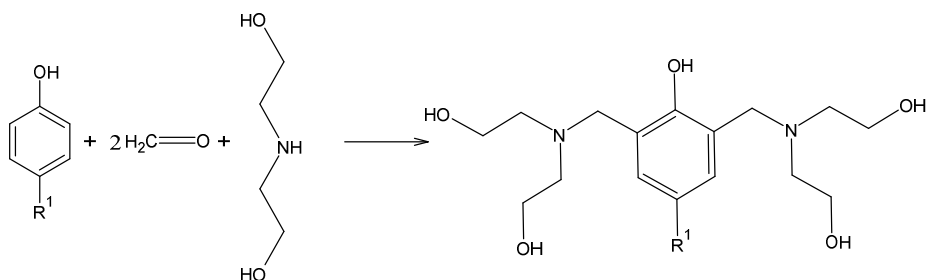
- poliole polieterowe,
- poliole poliestrowe.

Jednym z ważniejszych polieteroli do otrzymywania sztywnych pianek poliuretanowych, które spienia się wodą jest polioliol Mannicha. Są to związki, które zawierają w swojej budowie azot. Otrzymuje się je poprzez wcześniejsze przeprowadzenie reakcji pomiędzy fenolem, alkanoloaminami oraz formaldehydem, produktem jest zasada Mannicha, którą poddaje się alkoksylowaniu i dzięki temu powstaje polioliol Mannicha⁸.

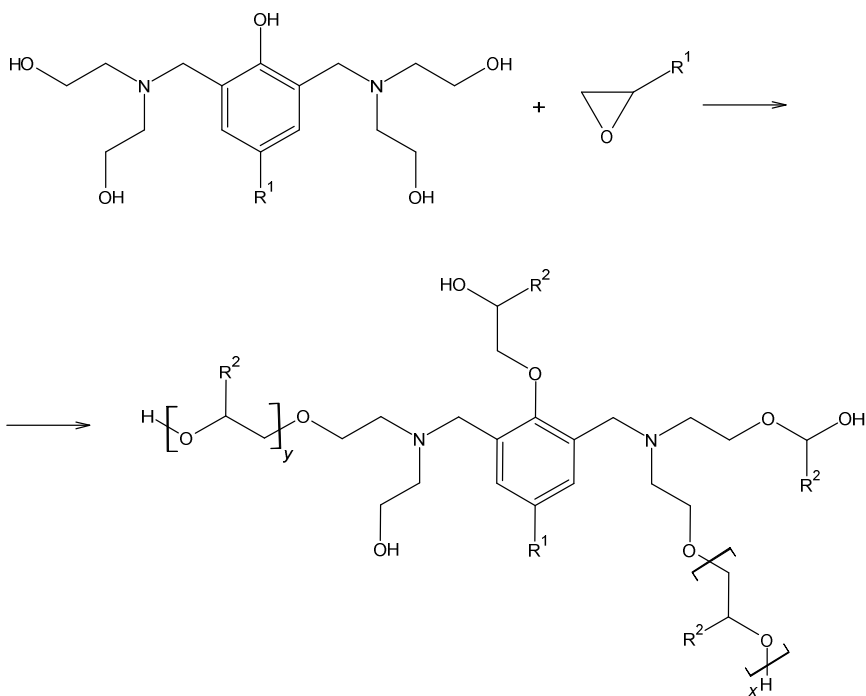
⁶ Zafar F., Sharmin E., *Polyurethane*, IntechOpen, Croatia, 2012, str.6-7

⁷ Parzuchowski P., Szymańska A., *Pianki poliuretanowe*
dostęp online 29.04.2022: http://pparzuch.ch.pw.edu.pl/ids/pianki_PUR.pdf

⁸ Nelson F. Molina, Stanley E. Moore, *Sposób wytwarzania polioli Mannicha*
dostęp online 29.04.2022: <https://patents.google.com/patent/PL204573B1/pl>



Rys. 1. Reakcja otrzymywania zasady Mannicha

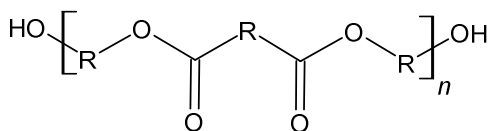


Rys. 2. Otrzymywanie polioliu Mannicha

Poliestrole są to małowcząsteczkowe poliestry, które jako pierwsze były wykorzystywane w roli polioliu do produkcji sztywnych pianek poliuretanowych. Otrzymuje się je na drodze reakcji polikondensacji, która zachodzi pomiędzy

wielofunkcyjnymi kwasami karboksylowymi a związkami polihydroksylowymi⁹.

Ogólny wzór poliesterolii przedstawiono poniżej:

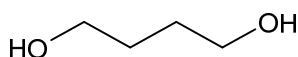


Rys. 3. Wzór ogólny poliesterolii

Małocząsteczkowe związki, powodujące wzrost ilości segmentów sztywnych, a zatem wzrost masy cząsteczkowej, nazywane są środkami przedłużającymi. Ich masa zawiera się w zakresie 40 - 300 Da. Nie różnią się one w znaczący sposób od środków sieciujących, ponieważ w trakcie reakcji następuje szybki wzrost łańcucha, a także powstawanie rozgałęzień. Bardziej rozbudowana budowa produktu zapewnia mu lepszą wytrzymałość mechaniczną, ale również odpowiada za stabilność termiczną i hydrolityczną¹⁰.

Jako przedłużacze stosuje się dwufunkcyjne związki takie jak diaminy czy diole. Związki wykazujące trój- lub większą funkcyjność należą do środków sieciujących.

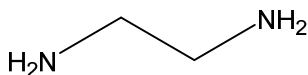
Do najważniejszych przedłużaczy łańcucha należą:



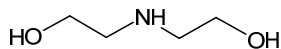
Butano-1,4-diol



Cykloheksano-1,4-diol



Etylenodiamina



Dietanoloamina

Bardzo ważnym reagentem w procesie otrzymywania pianek poliuretanowych jest czynnik gazotwórczy, powodujący powstanie struktury porowatej.

⁹ Ionescu M., *Chemistry and Technology of Polyols for Polyurethanes*, rapra TECHNOLOGY, UK 2005, str.1-10

¹⁰ Wirpsza Z., *Poliuretany. Chemia, technologia, i zastosowanie*, Wydawnictwo Naukowo-Techniczne, Warszawa 1991

Jednym z najłatwiej dostępnych poroforów jest woda. Wchodzi ona w reakcje z grupami izocyjanianowymi, w wyniku czego, podczas egzotermicznej reakcji, powstaje CO₂, dzięki któremu powstaje struktura komórkowa, a woda odparowuje.

Katalizatory są bardzo ważnymi związkami w syntezie poliuretanów. Odpowiadają one za zapewnienie odpowiedniej struktury produktu końcowego poprzez regulowanie względnej szybkości poszczególnych reakcji. Bardzo ważne jest dobranie odpowiedniego katalizatora, ponieważ zbyt szybki wzrost pianki może spowodować powstanie zbyt dużych porów, co jest niekorzystnym zjawiskiem jeśli chodzi o później wykonywane badania pod kątem np. wytrzymałości. Tak samo wzrost objętości nie może być zbyt wolny¹¹.

Istotnymi dodatkami, które były powodem przeprowadzonych badań są antypireny addytywne. Dzięki nim sztywne pianki poliuretanowe zyskują odporność na ogień. Celem artykułu jest przedstawienie ich wpływu również na właściwości fizyczne otrzymanych pianek. W tym celu wykonano badania gęstości pozornej, stabilności wymiarów, chłonności wody oraz współczynnika przewodzenia ciepła.

1. CZĘŚĆ DOŚWIADCZALNA

Pianki poliuretanowe otrzymano stosując komercyjny polioliol Rokopol RF 151V (PCC Rokita, Brzeg Dolny), MDI (Borsodchem, Hungary) oraz środki zmniejszające palność (melaminę, MP, MPP, izocyjanuran melaminy (ICM), Roflamy P, F5, B7L oraz exolit (Ex) i GE (w ilości 30% mas. w odniesieniu do masy polioliolu).

Oznaczono następujące właściwości fizyczne: gęstość pozorną (norma PN-EN ISO 845:2010), stabilność wymiarów (norma PN-92/C 89083), chłonność wody (norma PN-C-89084:1993), współczynnik przewodzenia ciepła (norma PN-EN 993-15:2006).

2. WYNIKI

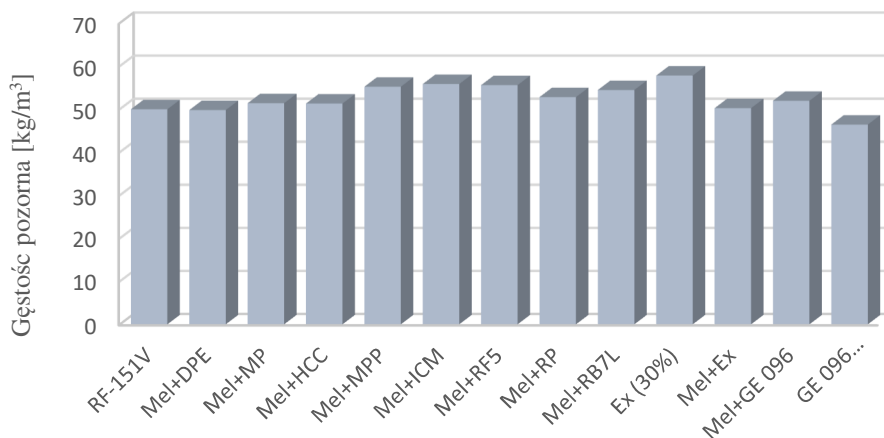
Gęstość pozorną sztywnych pianek poliuretanowych mieści się w zakresie 45-60 kg/m³, oznaczano korzystając z normy PN-EN ISO 845:2010, a uzyskane wyniki przedstawiono na wykresie (Wyk. I.).

Najmniejszą gęstość pozorną wykazuje pianka zawierająca GE 096 (30%); wynosi ona 46,4 kg/m³, co jest niższą wartością nawet od pianki referencyjnej RF-151V, dla której wartość to 49,9 kg/m³. Może być to spowodowane wysokim stopniem upakowania, a zatem najprawdopodobniej kompozycja ta zawiera w swojej strukturze mniej porów i są one mniejszej wielkości niż w przypadku

¹¹ Piszczyk Ł., Rozprawa doktorska. Synteza i właściwości nowych uniepalnionych pianek poliuretanowych otrzymanych z użyciem surowców odnawialnych, Politechnika Gdańska, Gdańsk, 2011

referencji. Największą gęstość pozorną zaobserwowano dla kompozycji z udziałem Exolitu ($57,8 \text{ kg/m}^3$).

Literaturowa gęstość pozorną dla sztywnych pianek poliuretanowych mieści się w zakresie od 28 do 60 kg/m^3 ¹².



Wyk. 1. Zestawienie gęstości pozornych otrzymanych pianek

Stabilność wymiarów oznaczano poprzez pomiar kształtek przed badaniem i po ich umieszczeniu w temperaturze 150°C na odpowiednio 20 i 40 godzin, a następnie obliczenie procentowej zmiany wymiarów liniowych.

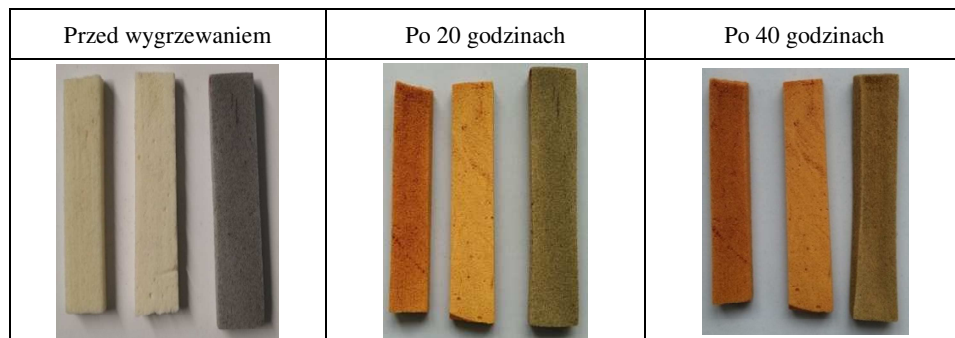


Fot. 1. Od lewej: RF-151V Ref, Ex 30%, GE096 30%.

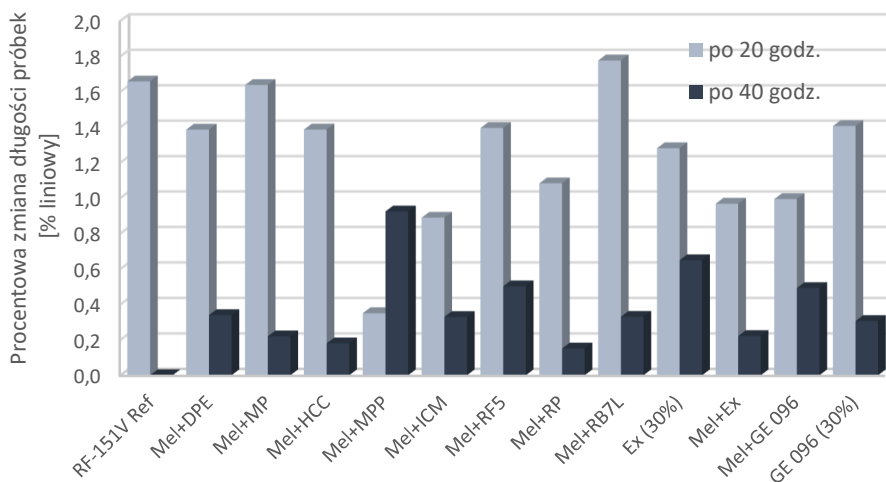
¹² Prociak A., Rokicki G., Ryszkowska J., *Materiały poliuretanowe*, Warszawa 2016, str. 166-167

Pianki poddano działaniu wysokiej temperatury (150°C) i obserwowano, jakie będzie to miało na nie wpływ. Zauważono nieznaczne zmiany w wymiarach, jednak porównując próbki przed, w trakcie i po zauważono różnice w ich wyglądzie. W tabeli przedstawiono pianki poddane badaniu (Tab.1).

Tab.1. Zestawienie próbek: RF-151V Ref, Ex (30%), GE 096 (30%).



Z obliczeń wynika, iż pianki po wygrzewaniu wykazują niewielki skurcz liniowy, tj. nieznaczne zmiany wymiarów, tj. długości, szerokości oraz wysokości. Procentową zmianę długości przedstawiono na poniższym wykresie (Wyk. 2.).



Wyk. 2. Zestawienie procentowej zmiany długości próbek po 20 oraz 40 godzinach.

Zmiany w wyglądzie kształtek zaobserwowano podczas pierwszych 20 godz. wygrzewania, po 40 godz. nie stwierdzono dalszych zmian.

Najmniejsze zmiany wymiarów stwierdzono w przypadku pianki referencyjnej, niewielkie natomiast w przypadku kompozycji Mel+MP, Mel+RP, Mel+Ex oraz GE 096 (30%).

W celu określenia chłonności wody otrzymanych kompozycji kształtki zmierzono i ważono, a następnie umieszczono w wodzie dejonizowanej w taki sposób, aby w całości znajdowały się pod jej powierzchnią. Kolejne pomiary wykonano odpowiednio po: 5 min., 3 i 24 godz. Na podstawie otrzymanych wyników wyznaczono objętościową chłonność wody.



Fot. 2. Próbkę przygotowane do pomiaru chłonności wody.

Badanie przeprowadzono w celu sprawdzenia w jakim stopniu otrzymane pianki są w stanie zaabsorbować wodę.

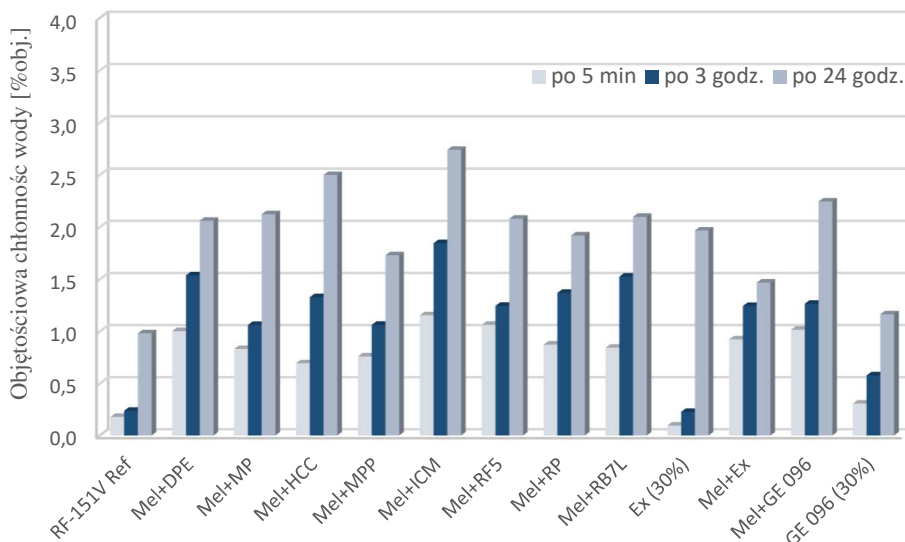
Tab. 2. Zestawienie wyników objętościowej chłonności wody

Lp.	Skrót kompozycji	Objętościowa chłonność wody [%obj.] po:		
		5 min.	3 godz.	24 godz.
1	RF-151V Ref	0,2	0,2	1,0
2	Mel+DPE*	1,0	1,5	2,1
3	Mel+MP*	0,8	1,1	2,1
4	Mel+HCC*	0,7	1,3	2,5
5	Mel+MPP*	0,8	1,1	1,7
6	Mel+ICM*	1,2	1,8	2,7
7	Mel+RF5*	1,1	1,2	2,1
8	Mel+RP*	0,9	1,4	1,9
9	Mel+RB7L*	0,8*	1,5	2,1
10	Ex (30%)	0,1	0,2	2,0
11	Mel+Ex*	0,9	1,2	1,5
12	Mel+GE 096*	1,0	1,3	2,2
13	GE 096 (30%)	0,3	0,6	1,2

*Dane do danej kompozycji zostały zaczerpnięte z inżynierskiej pracy Marty Sołtysik, *Uniepalniacze sztywnych pianek poliuretanowych*, Rzeszów 2020.

Wraz ze upływem czasu zanurzenia obserwuje się wzrost chłonności wody (Tab.2, Wyk.3).

Badanie objętościowej chłonności wody polega na obliczeniu objętości wody, którą zaabsorbowała pianka. Największą wartość po 24 godz. zanotowano dla kompozycji Mel+ICM (2,7% obj.), a kolejno dla Mel+HCC (2,5% obj.), i Mel+GE 096 (2,2% obj.).



Wyk. 3. Objętościowa chłonność wody po 5 min, 3 godz. oraz po 24 godz.

Najniższą wartość objętościowej chłonności wody zaobserwowano w przypadku pianki RF-151V Ref (1,0% obj.). Drugą najmniejszą chłonność objętościową uzyskała pianka z dodatkiem GE 096 (30%), (1,2% obj.). Z powyższych danych wynika, że dla wszystkich kompozycji wartości te mieszczą się w zakresie 1- 2,7% obj., zatem nie przekraczają one 3% obj. a literatura podaje, iż objętościowa chłonność wody nie powinna wynosić więcej niż 5% obj.¹³

Uzyskane wyniki wskazują na to, że otrzymane pianki wykazują odporność na działanie wody (1,0-2,7% obj.), jednak warto byłoby je zbadać w jeszcze dłuższym przedziale czasowym. Jak podaje firma SAAPPUR, chłonność wody poliuretanowych materiałów izolacyjnych nie powinna przekraczać 5% obj., gdyż wtedy nadają się do stosowania ich według faktycznego przeznaczenia¹⁴.

¹³ Radziszewska-Zielina E., Analiza porównawcza parametrów materiałów termoizolacyjnych mających zastosowanie jako izolacja ścian zewnętrznych, Przegląd budowlany, 4, 2009, 33

¹⁴ Produkty i firmy, informator budownictwa, *Pianka PIR, PUR, PE*, 2015

Do badania współczynnika przewodzenia ciepła wykorzystano aparat – IZOMET 2144 Applied Precision. Do wykonania pomiarów zastosowano pianki, do których wprowadzono sondę igłową, przez którą przepuszczono prąd o danym natężeniu.



Fot. 3. Pianki poddane badaniu współczynnika przewodzenia ciepła.
Od lewej: Mel+GE 096, GE 096 (30%).

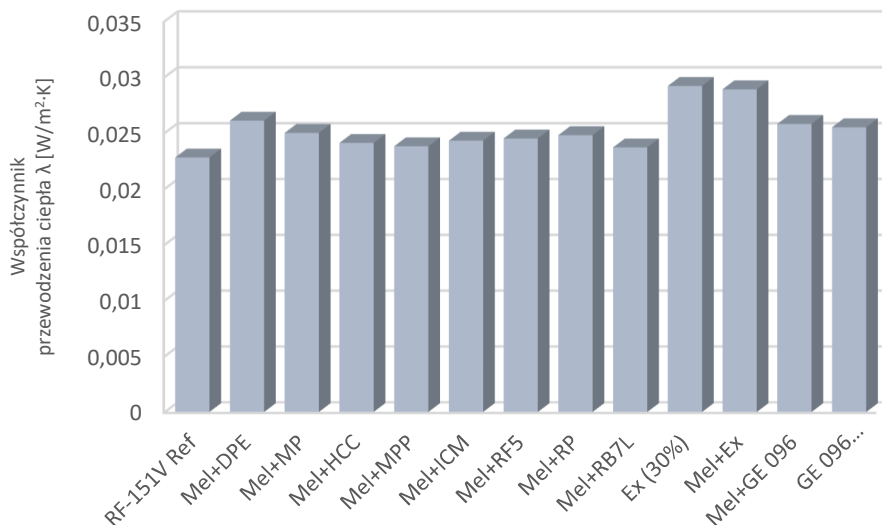
Współczynnik przewodzenia ciepła jest to ilość energii w postaci ciepła, która przepływa przez całość masy próbki, na skutek zewnętrznej różnicy temperatury. Im mniejsza jest ta wartość, tym dany materiał gorzej przewodzi ciepło, czyli jest lepszym izolatorem, czego oczekuje się od sztywnych pianek poliuretanowych.

Otrzymane wyniki umieszczone w tabeli (Tab.3.).

Tab.3. Wartości współczynników przewodzenia ciepła otrzymanych pianek.

Lp.	Skrót kompozycji	Współczynnik przewodzenia ciepła λ [W/m·K]
1.	RF-151V Ref*	0,0228 +/- 0,0072
2.	Mel+DPE*	0,0261 +/- 0,0027
3.	Mel+MP*	0,025 +/- 0,0026
4.	Mel+HCC*	0,0241 +/- 0,0032
5.	Mel+MPP*	0,0238 +/- 0,0015
6.	Mel+ICM*	0,0243 +/- 0,0023
7.	Mel+RF5*	0,0245 +/- 0,0013
8.	Mel+RP*	0,0248 +/- 0,0033
9.	Mel+RB7L*	0,0237 +/- 0,0054
10.	Ex (30%)*	0,0292 +/- 0,0042
11.	Mel+Ex*	0,0289 +/- 0,002
12.	Mel+GE 096	0,0258 +/- 0,002
13.	GE 096 (30%)	0,0255 +/- 0,002

Na podstawie tabeli stworzono wykres (Wyk.5.), z którego wynika, iż prawie wszystkie pianki otrzymane przy udziale antypirenów wykazują wyższy współczynnik przewodzenia ciepła niż pianka referencyjna (RF 151V Ref.).



Wyk. 5. Zestawienie współczynników przewodzenia ciepła.

Drugą najlepiej wypadającą próbką jest ta zawierająca Roflam B7L. Wyróżnia się sześć kompozycji, których wartość dla tego parametru przekroczyła lub jest równa 0,025 W/m·K. W kolejności malejącej są to: Ex (30%) (0,0292 W/m·K), Mel+Ex (0,0289 W/m·K), Mel+DPE (0,0261 W/m·K), Mel+GE 096 (0,0258 W/m·K), GE 096 (30%) (0,0255 W/m·K) oraz Mel+MP (0,025 W/m·K). Jak widać, dodatek Exolitu OP 935 negatywnie wpływa na właściwości izolacyjne sztywnych pianek poliuretanowych, ponieważ obie kompozycje z jego udziałem uzyskały największe wartości współczynnika przewodzenia ciepła, a wspomniano już, że powinna być ona jak najmniejsza. Dla przykładu, komercyjne pianki posiadają wartości współczynnika przewodzenia ciepła w zakresie od 0,023 do 0,035 W/m·K¹⁷. Porównując dane uzyskane z 13 otrzymanych kompozycji można wyróżnić 7 spełniających podany warunek.

3. PODSUMOWANIE

Na podstawie przeprowadzonych badań stwierdzono, że wyznaczone wartości gęstości pozornych mieszczą się w granicach literaturowych wynoszących

28-60 kg/m³. Wszystkie kompozycje wykazują niższy współczynnik przewodzenia ciepła (0,0228–0,0292 W/m·K) i chłonność wody (poniżej 3% obj.) niż pianki stosowane do ocieplania (0,023 do 0,035 Wm·K¹⁷, 5% obj.¹⁷. Żadna z kompozycji (poddana działaniu temperatury wynoszącej 150°C), nie wykazała zmiany wymiarów liniowych przekraczających 1%. Uzyskane wyniki pozwalają przypuszczać, iż otrzymane pianki poliuretanowe o zwiększonej odporności na ogień będzie można zastosować jako bezpieczne materiały izolacyjne.

LITERATURA

1. Ionescu M., Chemistry and Technology of Polyols for Polyurethanes, rapra TECHNOLOGY, UK 2005
2. Kricheldorf H.R., Nuyken O., Swift G., *Handbook of Polymer Synthesis second edition*. CRC Press, Nowy Jork, USA 2004, str. 503-505
3. Piszczuk Ł., Rozprawa doktorska. Synteza i właściwości nowych uniepalnionych pianek poliuretanowych otrzymywanych z użyciem surowców odnawialnych, Politechnika Gdańska, Gdańsk, 2011
4. Pivec T., Smole M. Sfiligoj, Gašparič P., Stana Kleinschek K., *Polyurethanes for Medical Use*, Tekstilec, 60 (4), 2017
5. Prociak A., Rokicki G., Ryszkowska J., *Materiały poliuretanowe*, Warszawa 2016
6. Radziszewska-Zielina E., Analiza porównawcza parametrów materiałów termoizolacyjnych mających zastosowanie jako izolacja ścian zewnętrznych, *Przegląd budowlany*, 4, 2009, 33
7. Sardon H., Pascual A., Mecerreyes D., Taton D., Cramail H., Hedrick J. L., *Synthesis of Polyurethanes Using Organocatalysis: A Perspective*, *Macromolecules*, 48 (10), 2015
8. Szycher M., *Shycher's handbook of polyurethanes*, CRC Press LLC, New York 1999, str. 1-2,
9. Wirpsza Z., *Poliuretany. Chemia, technologia, i zastosowanie*, Wydawnictwo Naukowo-Techniczne, Warszawa 1991
10. Zafar F., Sharmin E., *Polyurethane*, IntechOpen, Croatia, 2012
11. Zwi A.B., *Isocyanates and health – a review*, *South African Medical Journal*, 67, 1985

ŹRÓDŁA INTERNETOWE

12. Parzuchowski P., Szymańska A., *Pianki poliuretanowe*, Dostęp online 17.01.2022: http://pparzuch.ch.pw.edu.pl/ids/pianki_PUR.pdf
13. Produkty i firmy, informator budownictwa, *Pianka PIR, PUR, PE*, 2015, Dostęp online: <https://budujemydom.pl/irbj/aktualnosci/44388-wszystkiego-o-piance-poliuretanowej-dowiedz-sie-od-firmy-saappur>
14. Nelson F. Molina, Stanley E. Moore, *Sposób wytwarzania polioli Mannicha*, Dostęp online 29.01.2010 : <https://patents.google.com/patent/PL204573B1/pl>

Ewelina SZOSTEK

dr inż. Dorota GŁOWACZ-CZERWONKA, prof. PRz
opiekun naukowy

BADANIE PALNOŚCI PIANEK POLIURETANOWYCH NA BAZIE ROKOPOLU RF-551 Z DODATKIEM WYBRANYCH UNIEPALNIACZY ADDYTYWNYCH

Otrzymano sztywne pianki poliuretanowe (oparte na Rokopolu RF 551) z dodatkiem wybranych uniepalniaczy addytywnych, a następnie zbadano właściwości pożarowe otrzymanych kompozycji. Wyniki odniesiono do pianki referencyjnej. Celem przeprowadzonych badań było otrzymanie pianek o jak najwyższych właściwościach pożarowych. Aby tego dokonać przeprowadzono dwa badania: metoda indeksu tlenowego oraz test poziomy. Badanie pierwsze wykazało, że najwyższe wartości LOI otrzymano dla kompozycji melamina z Exolitem. Test poziomy wskazał również dobre wyniki dla kompozycji z samym Grafitem i samym Exolitem. Wskazane jest kontynuowanie badań związanych z wpływem antypirenów na uniepalnianie pianek poliuretanowych.

Słowa kluczowe: sztywne pianki poliuretanowe, palność, antypiren addytywny, polioliol

WPROWADZENIE

Poliuretany (PUR) to tworzywa sztuczne z grupy polimerów, które swoją popularność zawdzięczają możliwości wytwarzania ich wykorzystując różne metody co w efekcie daje szeroką gamę produktów o bardzo zróżnicowanych właściwościach.¹ Materiał ten może być wykorzystywany w postaci pianek, lakierów, elastomerów, klejów czy włókien. Sztywne pianki poliuretanowe (SPPUR) stanowią bardzo liczną grupę materiałów na produkcję, których zużywa surowce PUR.¹ Są one zwykle wykorzystywane do termoizolacji, dlatego też przy ich wytwarzaniu zwraca się szczególną uwagę na takie parametry jak struk-

¹ Prociak A., Rokicki G., Ryszkowska J., *Materiały poliuretanowe*, WN PWN, Warszawa 2016, 11-16, 162-164

tura zamkniętokomórkowa, odporność na starzenie, niskie przewodnictwo ciepłe oraz dobra stabilność wymiarów podczas ekspozycji na wysokie temperatury. ¹ Dużym problemem tych materiałów jest ich palność, której towarzyszy wydzielanie się toksycznych gazów. SPPUR wspomagają spalanie i palą się dość szybko ze względu na niski indeks tlenowy (LOI).

Wymaga się więc aby były ognioodporne do różnych zastosowań. Można tego dokonać poprzez wprowadzenie w ich struktury substancji zmniejszających palność – niepalniaczy. Pianki zawierające odpowiednio dobrane środki niepalające wykazują właściwości samogasnące, czyli nie będą powodować dalszego rozprzestrzeniania się pożaru. Pianka ulegnie samoistnemu ugaszeniu po ustąpieniu źródła ognia. Działanie to może być spowodowane np. przez: wytworzeniem warstwy ochronnej, która utrudnia dopływ ciepła do wnętrza pianki.²

Poliuretany (PU) to materiały zawierające grupę uretanową – skąd też pochodzi ich nazwa. Są one otrzymywane w reakcji polioliu i diizocyjanianu w reakcji polimeryzacji zwanej poliaddycją.³ Można śmiało stwierdzić, że materiały te zmieniły świat. Są one bardzo różnorodną grupą związków i dlatego nadają się do szerokiego zakresu zastosowań w różnych sektorach przemysłu i produkcji, takich jak motoryzacja, budownictwo, elektronika, meble czy przemysł opakowaniowy. Zawdzięczamy im farby, lakiery, energooszczędne lodówki, wygodne meble tapicerowane, panele izolacyjne, elastomerowe koła i opony czy lekkie kompozyty.¹

Obecnie poliuretany zajmują piąte miejsce pod względem produkcji jak i zastosowania polimerów. Wielkość rynku poliuretanów na całym świecie w latach 2015–2020 utrzymywała się na poziomie ponad 21 mln ton, przy czym z roku na rok ta wartość była coraz większa w 2019 wyniosła 25,1 mln ton. W roku 2020 produkcja poliuretanu odnotowała delikatny spadek w porównaniu do roku poprzedzającego i wyniosła 23,9 mln ton. Spadek ten jest spowodowany epidemią COVID-19, której skutki były widoczne w każdej branży na całym świecie. Rozwój rynku poliuretanów był utrudniony z takich powodów jak: ograniczenia rządowe, ograniczenia w podróżowaniu, blokady i zamykanie fabryk. Ponieważ przemysł budowlany został wstrzymany, miało to negatywny wpływ na rynek poliuretanu. Również w przemyśle samochodowym, różne zakłady produkujące samochody i komponenty tymczasowo zamknęły swoją produkcję i działalność biurową, co doprowadziło do gwałtownego spadku sprzedaży samochodów, a co za tym idzie – zmniejszyło się zapotrzebowanie na poliuretany.⁴ Prognozy na lata 2021-2026 wskazują, że rynek tych materiałów będzie

² Janowska G., Przygocki W., Włochowicz A., *Palność polimerów i materiałów polimerowych*, WNT, Warszawa 2007, 183-204

³ https://www.chemistryviews.org/details/ezone/10695960/80_Years_of_Polyurethane.html (dostęp: 16.10.2021)

⁴ Mordor Intelligence. Polyurethane Market – Growth, Trends, COVID-19 Impact, and Forecasts (2021-2026) 2020. 1-15

utrzymywał tendencję wzrostową. Szacuje się, że wielkość światowego rynku poliuretanów wzrośnie do 27,6 mln ton w roku 2026.⁵

Tak wielkie zainteresowanie poliuretany zawdzięczają swoim wszechstronnym właściwościom, które są uwarunkowane przez takie czynniki jak *m.in.*: rodzaje użytych surowców, ich wzajemny stosunek, dodatek substancji zmieniających właściwości czy metoda otrzymania polimeru. Sterując tymi parametrami możliwe jest wytworzenie różnego typu materiałów poliuretanowych. Poliuretany można podzielić na kilka zasadniczych typów produktów takich jak: pianki, elastomery, farby, lakiery, kleje, materiały do powlekania, włókna a także środki nadające elastyczność innym produktom.

Największe znaczenie przemysłowe zyskały pianki poliuretanowe, które można rozdzielić na trzy zasadnicze grupy: pianki elastyczne (miękkie), sztywne (twarde) oraz półsztywne. Różnią się one od siebie właściwościami fizykochemicznymi. Mimo, że otrzymywanie pianek elastycznych i sztywnych odbywa się na podobnej zasadzie chemicznej, to różnice w ich właściwościach wynikają z odmiennej charakterystyki reagentów, a mianowicie użytych polioli i izocyjanianów.⁶ Materiały te charakteryzują się lekkością oraz dobrymi parametrami termicznymi, izolacyjnymi i akustycznymi. Są one odporne na grzyby i pleśń oraz stabilne, co oznacza, że wraz z upływem czasu nie zmieniają swoich parametrów mechanicznych oraz izolacyjnych. Pianki są łatwe w obróbce i w montażu, a przede wszystkim nie są szkodliwe dla zdrowia. Materiały te są liderami w dziedzinie szeroko rozumianego komfortu. Wytwarza się z nich między innymi materace, poduszki, siedziska czy elementy do krzeseł i foteli. Stosuje się je także w przemyśle samochodowym, tekstylnym czy obuwniczym. Ogromną popularność tworzywo to zyskało jednak w budownictwie jako materiał do montażu, wygłuszenia ścian działowych oraz uszczelniania okien czy drzwi.⁷

1. CZĘŚĆ DOŚWIADCZALNA

Sztywne pianki poliuretanowe (SPPUR) otrzymywano w wyniku polimeryzacji polioli Rokopol RF-551 z diizocyjanianem (MDI). Do pojemnika z tworzywa sztucznego z płaskim dnem odmierzone odpowiednie ilości następujących substratów: polioli (Rokopol RF-551), środek powierzchniowo czynny, katalizator, wodę dejonizowaną, wybrane uniepalniacze oraz MDI. Składniki mieszano do momentu kiedy jej konsystencja zaczęła przypominać krem. Obserwowano wzrost pianki zapisując czasy: wzrostu, kremowania i schnięcia. Po upływie 72 godzin wycinano kształtki potrzebne do zbadania ich właściwości pożarowych.

⁵ <https://www.statista.com/statistics/720341/global-polyurethane-market-size-forecast/> (dostęp: 16.10.2021)

⁶ N. V. Gama, A. Ferreira, A. Barros-Timmons *Polyurethane Foams: Past, Present, and Future*. Materials 2018, 11, 1-4

⁷ <https://muratorodom.pl/budowa/dach/pianka-poliuretanowa-najlepszy-termoizolator-domu-poradnik-aa-JbVS-1ZJ6-NdGQ.html> (dostęp na dzień 16.10.2021)

Zadano wybrane właściwości pożarowe SPUR, tj. indeks tlenowy i test poziomy. Badanie indeksu tlenowego przeprowadzono w oparciu o normę⁸ oraz korzystając z aparatury Concept Fire Testing. Natomiast test palności poziomej wykonano według normy⁹ przy użyciu metalowej łapy na stojaku oraz palnika.

2. WYNIKI BADAŃ WŁAŚCIWOŚCI POŻAROWYCH SPPUR

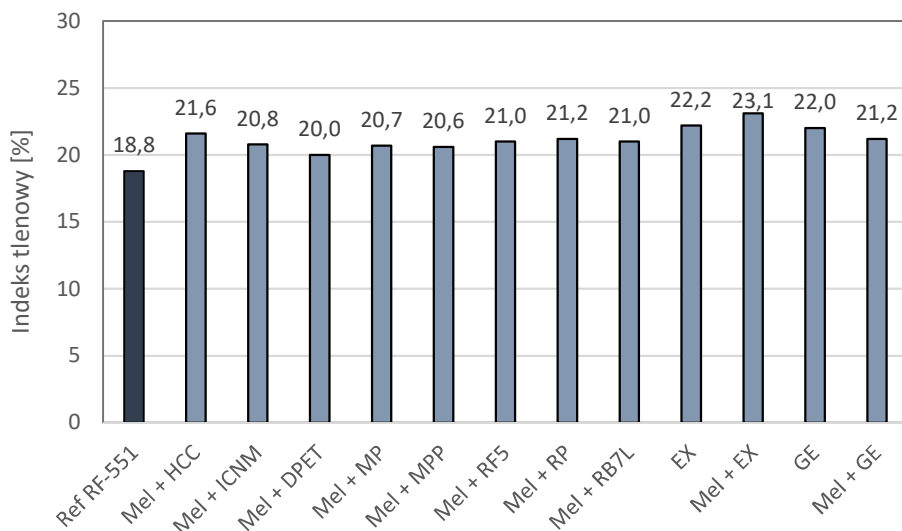
Badanie indeksu tlenowego wykazało, że w wyniku wprowadzenia uniepalniaczy otrzymano pianki samogasnące. Wartość LOI dla pianki referencyjnej wynosi 18,8 % obj. tlenu. Tak niska wartość spowodowana jest brakiem substancji zmniejszających palność. Użyty polioliol – Rokopol RF-551 dodatkowo przyczynia się do niskiej wartości indeksu tlenowego, z tego względu, że jego starterem jest sorbitol, a cukry same w sobie są palne. Najlepszą wartość uzyskano dla kompozycji: melamina z Exolitem (23,1% obj. tlenu). Kilka kompozycji osiąga wartości, które klasyfikują je jako substancje palne a są to: Mel + ICNM, Mel + DPET, Mel + MP oraz Mel + MPP. W Tabeli 1. zestawione zostały wartości LOI dla każdej otrzymanej kompozycji SPPUR. Na wykresie słupkowym (Rysunek 1.) przedstawione zostały indeksy tlenowe dla poszczególnych kompozycji piankowych.

Tabela 1. Zestawienie wyników badania indeksu tlenowego

Lp.	Nazwa kompozycji	Indeks tlenowy [% obj. O ₂]	Klasyfikacja palności
1	Ref RF-551	18,8	palna
2	Mel + HCC	21,6	samogasnąca
3	Mel + ICNM	20,8	palna
4	Mel + DPET	20,0	palna
5	Mel + MP	20,7	palna
6	Mel + MPP	20,6	palna
7	Mel + RF5	21,0	samogasnąca
8	Mel + RP	21,2	samogasnąca
9	Mel + RB7L	21,0	samogasnąca
10	Ex	22,2	samogasnąca
11	Mel + Ex	23,1	samogasnąca
12	GE	22,0	samogasnąca
13	Mel + GE	21,2	samogasnąca

⁸ PN-EN ISO 4589-3:2017-06 Tworzywa sztuczne - Oznaczanie zapalności metodą wskaźnika tlenowego, PKN Warszawa

⁹ PN-EN ISO 3582:2002 Elastyczne tworzywa sztuczne porowate - Ocena laboratoryjna charakterystyk palenia małych próbek spalanych małym płomieniem w położeniu poziomym, PKN, Warszawa



Rysunek. 1. Indeks tlenowy dla otrzymanych SPPUR

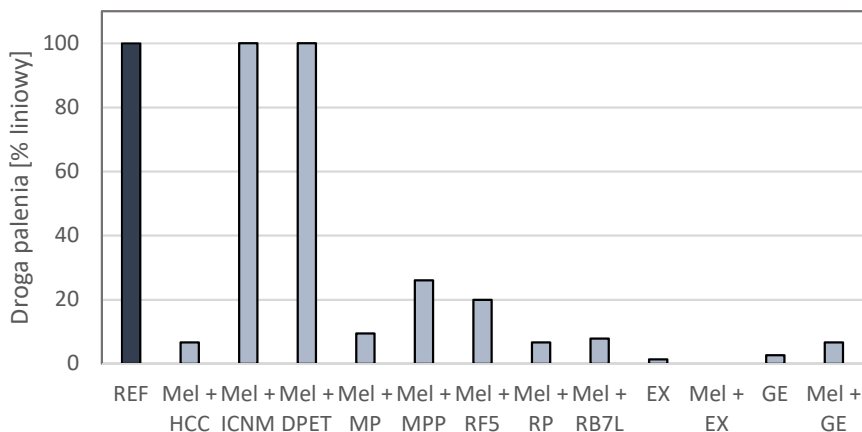
W teście poziomym badano drogę palenia, czas palenia, szybkość palenia oraz ubytek masy kompozycji SPPUR. Wyniki badań przedstawiono w Tabeli 2.

Tabela. 2. Zestawienie wyników badania testu poziomego

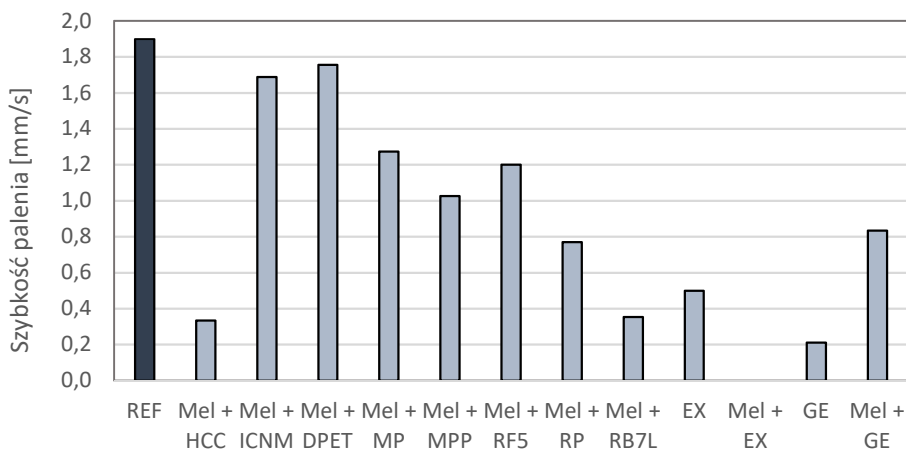
Lp.	Nazwa kompozycji	Droga palenia [% liniowy]	Czas palenia [s]	Szybkość palenia [mm/s]	Ubytek masy [% mas.]
1	Ref	100	79	1,90	65,08
2	Mel + HCC	6,7*	30*	0,33*	3,84
3	Mel + ICNM	100*	90*	1,69*	65,19
4	Mel + DPET	100*	86*	1,76*	63,52
5	Mel + MP	9,5*	11*	1,27*	5,50
6	Mel + MPP	26,0*	38*	1,03*	16,46
7	Mel + RF5	20,0*	25*	1,20*	12,13
8	Mel + RP	6,7*	13*	0,77*	3,53
9	Mel + RB7L	7,9*	34*	0,35*	3,56
10	Ex	1,3	4	0,50	1,44
11	Mel + Ex	0*	0*	0*	0,20
12	GE	2,7	19	0,21	0,31
13.	Mel + GE	6,7*	12*	0,83*	4,16

* Dane z pracy mgr. (2021) oraz inż. (2020) Magdaleny Sajek. Niniejsze badania przedstawione w artykule to uzupełnienie brakujących danych z powyżej wymienionych prac.

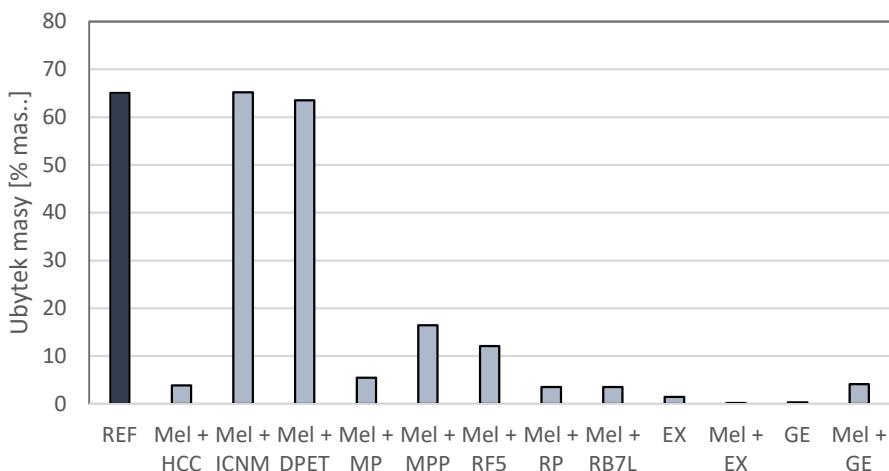
Rys. 2 do Rys. 4 przedstawiają te wartości na wykresach słupkowych. Dla pianki referencyjnej droga palenia wynosi 100% liniowych długości pianki, a szybkość jej palenia jest większa od wszystkich pozostałych kompozycji. Podobne wartości uzyskano w wypadku kompozycji: Mel+ICNM oraz Mel+DPET. Pianki te uległy całkowitemu spaleniu.



Rysunek. 2. Droga palenia dla otrzymanych SPPUR



Rysunek. 3. Szybkość palenia dla otrzymanych SPPUR



Rysunek. 4. Ubytek masy dla otrzymanych SPPUR

W wypadku Mel + MPP przekroczona została linia graniczna, a droga palenia osiągnęła 26% liniowych, nie spaliła się jednak cała pianka, w odróżnieniu od w/w kompozycji. W przypadku innych pianek płomień nie przekroczył linii. Najlepiej zachowała się pianka Mel+Ex, która nie ulegała zapaleniu, była jedynie osmalona.

PODSUMOWANIE

Otrzymano sztywne pianki poliuretanowe na bazie polioliu Rokopol RF-551 z dodatkiem uniepalniaczy, a następnie zbadano ich właściwości pożarowe. W wyniku wprowadzenia uniepalniaczy otrzymano pianki samogasnące, czyli takie z LOI większym niż 21% obj. tlenu. Najwyższą wartość uzyskano dla kompozycji: Mel +Ex (LOI =23,1). Test poziomy wykazał, że najlepsze właściwości pożarowe wykazują kompozycje: Mel + Ex oraz pianka z samym GE i samym Ex.

Badanie to wykazało również, że nie warto kontynuować badań w oparciu o HCC oraz DPET. Wskazane jest kontynuowanie badań związanych z wpływem antypirenów na uniepalnianie pianek poliuretanowych.

LITERATURA

1. Prociak A., Rokicki G., Ryszkowska J., *Materiały poliuretanowe*, WN PWN, Warszawa 2016, 11-16, 162-164
2. Janowska G., Przygocki W., Włochowicz A., *Palność polimerów i materiałów polimerowych*, WNT, Warszawa 2007, 183-204
3. Mordor Intelligence. Polyurethane Market – Growth, Trends, COVID-19 Impact, and Forecasts (2021-2026) 2020. 1-15
4. N. V. Gama, A. Ferreira, A. Barros-Timmons *Polyurethane Foams: Past, Present, and Future*. Materials 2018, 11, 1-4

AKTY NORMATYWNE

5. PN-EN ISO 4589-3:2017-06 Tworzywa sztuczne – Oznaczanie zapalności metodą wskaźnika tlenowego, Polski Komitet Normalizacyjny, Warszawa
6. PN-EN ISO 3582:2002 Elastyczne tworzywa sztuczne porowate - Ocena laboratoryjna charakterystyk palenia małych próbek spalanych małym płomieniem w położeniu poziomym, Polski Komitet Normalizacyjny, Warszawa

ŹRÓDŁA INTERNETOWE

7. https://www.chemistryviews.org/details/ezine/10695960/80_Years_of_Polyurethane.html (dostęp: 16.10.2021)
8. <https://www.statista.com/statistics/720341/global-polyurethane-market-size-forecast/> (dostęp: 16.10.2021)
9. <https://murator-dom.pl/budowa/dach/pianka-poliuretanowa-najlepszy-termoizolator-domu-poradnik-aa-JbVS-1ZJ6-NdGQ.html> (dostęp na dzień 16.10.2021)

Ewelina SZOSTEK

dr inż. Dorota GŁOWACZ-CZERWONKA, prof. PRz
opiekun naukowy

ANTYPIRENY I ICH ROLA W UNIEPALNIANU PIANEK POLIURETANOWYCH

Sporym problemem pianek poliuretanowych jest ich palność, której dodatkowo towarzyszy wydzielanie się toksycznych gazów. Dodatkowo wspomagają spalanie i palą się dość szybko ze względu na niski indeks tlenowy (LOI). Aby SPPUR były ognioodporne do różnych zastosowań wprowadza się w ich struktury substancje zmniejszające palność czyli – uniepalniacze. Pianki zawierające odpowiednio dobrane środki uniepalniające wykazują właściwości samogasnące, czyli ulegają samoistnemu ugaszeniu po ustąpieniu źródła ognia. W poniższym artykule skupiono się na zjawisku palności pianek poliuretanowych oraz na ogólnej charakterystyce środków mających na zadanie zmniejszenie albo nawet wyeliminowanie tego zjawiska. Celem artykułu jest przedstawienie zasady działania uniepalniaczy oraz charakterystyka wybranych związków należących do tej grupy ale lepiej zrozumieć działanie tych substancji.

Słowa kluczowe: palność, antypiren addytywny, polirole, uniepalnianie

WPROWADZENIE

Palenie jest to szybki, samorzutny i samopodtrzymujący się proces utleniania – redukcji, któremu towarzyszy tworzenie się płomienia oraz wydzielanie się ciepła. W spalaniu bierze udział paliwo oraz tlen. Paliwem jest każda substancja, która w reakcji z tlenem (utleniaczem) wydziela ciepło. Do paliw można zaliczyć np. gaz ziemny, tłuszcze roślinne, węgiel kamienny, koks, drewno, związki organiczne (a w szczególności polimery syntetyczne).¹

Palność materiałów oznacza ich podatność na spalanie płomieniowe, przebiegające w palnej mieszance gazowej, zdolność materiału do podtrzymywania ognia. Tworzywa oparte na większości polimerów odznaczają się większym lub mniejszym stopniem palności. Polimery syntetyczne są bardziej łatwopalne niż

¹ Janowska G., Przygocki W., Włochowicz A., *Palność polimerów i materiałów polimerowych*, WNT, Warszawa 2007, 25-26, 187, 199-204.

np. polietylen czy polistyren. Mają one właściwości opałowe porównywalne z ropą naftową. W przeciwieństwie do polimerów naturalnych, które mogą być opóźnione w rozprzestrzenianiu się ognia jedynie poprzez powlekanie lub inną obróbkę powierzchni, proces produkcyjny większości tworzyw sztucznych jest idealny do zastosowań środków zmniejszających palność.²

Sposoby otrzymywania polimerów o zwiększonej odporności na ogień:

1. synteza niepalnych polimerów
2. zastosowanie niepalniaczy
3. chemiczna ingerencja na strukturę polimerów
4. zastosowanie powłok odpornych na ogień

Poliuretany pozbawione środków zmniejszających palność zalicza się do materiałów łatwopalnych. W szczególności mowa tutaj o wyrobach piankowych, ponieważ posiadają bardzo rozwiniętą powierzchnię porów ułatwiających spalanie. Poliuretany ulegają spalaniu w temperaturze przekraczającej 200°C. Spalaniu PUR towarzyszy wydzielanie toksycznych gazów.

Palność pianek PUR zależy będzie od wielu czynników takich jak m.in. skład chemiczny pianki, jej kształt, objętość, porowatość i inne. Są więc one łatwo zapalne i wysoce łatwopalne, wspomagają spalanie i palą się dość szybko. W związku z tym wymaga się, aby były ognioodporne dla różnych zastosowań. Pianki poliuretanowe podczas spalania wydzielają bardzo dużo dymu zaciemniającego wzrok. Toksyczność produktów spalania jest znacznie wyższa niż w przypadku wielu innych polimerów wytworzonych przez człowieka, ze względu na wysokie stężenia cyjanowodoru i tlenku węgla. Pianki PU stały się ognioodporne przez wprowadzenie związków zawierających fosfor, halogen, azot i różnych dodatków organicznych i nieorganicznych. Dodatki zmniejszające palność zmniejszają palność, poziom zadymienia i toksyczność pianek poliuretanowych z pewną degradacją innych właściwości.³

Celem artykułu jest przedstawienie zasady działania niepalniaczy, przybliżenie metod badania palności oraz charakterystyka wybranych niepalniaczy.

1. METODY BADANIA PALNOŚCI PIANEK POLIURETANOWYCH

Jedną z najczęściej stosowanych metod badania palności pianek poliuretanowych jest metoda indeksu tlenowego (IO). Polega ona na spalaniu próbki materiału o określonych wymiarach w kontrolowanej atmosferze, w której azot i tlen jest zmieszany w odpowiednich proporcjach. Metoda ta jest bardzo dokładna.

² Hull T. R., Stec A. A., *Polymers and fire*, Fire Retardancy of Polymers: New Strategies and Mechanisms, Royal Society of Chemistry, 2009, 1-14

³ Singh H., Jain A.K., Ignition, combustion, toxicity, and fire retardancy of polyurethane foams: A comprehensive review. *Journal of Applied Polymer Science*, 10, 2008, 1-9

Wartość IO wyrażana jest w procentach. Indeks tlenowy jest to więc najmniejsze stężenie tlenu, w mieszaninie tlenu i azotu, w którym dla danego materiału jest podtrzymywane spalanie. Im wyższa wartość IO, tym mniejsza skłonność materiału do palenia się, ponieważ potrzebuje on większej ilości tlenu do tego procesu. Do klasyfikacji palności materiałów (*na podstawie wartości IO*) stosuje się następujący podział:¹ materiał łatwopalny – IO poniżej 21% obj. O₂, materiał samogasnący – IO w przedziale 21 – 28% oraz materiał trudnopalny – IO powyżej 28% . Wartość ta zależy od takich czynników jak m.in. kształt, masa i grubość badanego materiału, rodzaj pianki czy zawartość wilgoci w piance. Wartość IO zależy również od czynników zewnętrznych np. ciśnienie, temperatura zapłonu czy sposób zainicjowania spalania.

Kalorymetria stożkowa to metoda ta polegająca na wykorzystaniu kalorymetru stożkowego, czyli urządzenia służącego do podpalenia badanego materiału i pomiaru ilości zużytego tlenu przez próbkę w czasie spalania. Aparat gromadzi dane dotyczące czasu zapłonu, ubytku masy, szybkości wydzielania ciepła i innych parametrów związanych z właściwościami spalania próbki. Niektóre kalorymetry stożkowe dają możliwość wyznaczenia toksyczności produktów rozkładu spalania takich jak np. CO, a najnowsze urządzenia wyposażone są nawet w spektrometry FTIR.¹ Zasada pomiaru wydzielania ciepła opiera się na zależności mówiącej, że ciepło spalania jest proporcjonalne do zużytego podczas spalania tlenu.

Metody badania palności za pomocą testu poziomego i testu pionowego określają łatwość zapłonu materiałów, szybkość ich spalania oraz reakcję na ogień. Materiały umieszczane są nad palnikiem Bunsena w pozycji pionowej (Vertical Testing – VT) lub poziomej (Horizontal Testing – HB). Podczas pionowej próby palności, materiał jest obserwowany pod kątem długości czasu palenia się po usunięciu płomienia, jak duża część próbki się pali i czy nie kapią z niej płonące cząsteczki. W poziomej próbie palności obserwuje się natomiast, czy materiał nadal się pali po usunięciu płomienia, a następnie oblicza się szybkość spalania próbki.⁴

2. UNIEPALNIACZE

Obecnie najczęściej stosowaną metodą uniepalenia polimerów jest stosowanie uniepalniaczy. Środki zmniejszające palność to związki chemiczne, które są dodawane do materiałów w celu zmniejszenia ich palności. Ma to na celu zmniejszenie prawdopodobieństwa wybuchu pożaru oraz zmniejszenie szybkości szerzenia się płomienia. Ich działanie polega w zależności od rodzaju użytego uniepalniacza na: wytworzeniu warstwy ochronnej utrudniającej dopływ ciepła do wnętrza tworzywa, obniżeniu temperatury płomienia oraz/lub na wy-

⁴ <https://www.element.com/materials-testing-services/flammability-test-methods> (dostęp na dzień 10.11.2021)

tworzeniu podczas spalania i topienia się tworzywa powłoki działającej jak warstwa izolująca palący się materiał od powietrza.¹

Antypireny dzielą się na addytywne i reaktywne. Te pierwsze stosowane są w trakcie kształtowania materiału polimerowego, pełnią również rolę napełniaczy czy plastyfikatorów. Natomiast antypireny reaktywne są wbudowywane w strukturę cząsteczki w trakcie jej syntezy.⁵

Właściwy dobór środków zmniejszających palność nie jest łatwy ze względu na bardzo dużą ilość różnych materiałów polimerowych, które różnią się między sobą pod względem budowy i składu. Przy wyborze ilości i rodzaju środka zmniejszającego palność należy również brać pod uwagę przeznaczenie wyrobu oraz stopień zagrożenia pożarowego.

Wśród środków zmniejszających palność wyróżnia się:

1. chlorowane i bromowane środki zmniejszające palność
2. fosforowe środki zmniejszające palność
3. środki zmniejszające palność na bazie azotu
4. nieorganiczne środki zmniejszające palność i związki mineralne.

Różne związki nieorganiczne i mineralne są łączone z bromem, fosforem lub azotem i stosowane jako środki uniepalniające lub jako elementy systemów uniepalniających. Związki nieorganiczne obejmują związki na bazie azotu, grafitu, krzemionki i nieorganicznych fosforanów. Związki mineralne obejmują niektóre fosforany, tlenki metali, wodorotlenki i inne produkty metalowe, takie jak glin, cynk i magnez.

Związki nieorganiczne i mineralne stosowane wraz z innymi pierwiastkami mogą pomóc w osiągnięciu bezpieczeństwa pożarowego w piankach poliuretanowych.⁶

Każdy uniepalniacz wprowadzony do pianki poliuretanowej wywołuje zmianę wielu właściwości np. mechanicznych, fizycznych, barwy.

Celem badań naukowych jest więc uzyskanie jak najlepszych właściwości pianki, przy równoczesnym zminimalizowaniu kosztów i maksymalizacji ognioodporności. Można stwierdzić, że mimo wielu znaczących prób, nie jest dostępne komercyjne rozwiązanie uniepalniania pianek poliuretanowych bez pewnej utraty właściwości fizycznych i mechanicznych.³

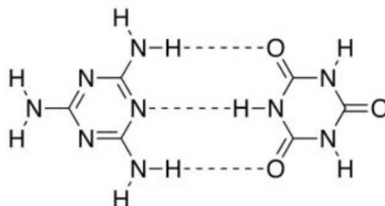
W literaturze istnieją pewne sprzeczności co do wpływu środków opóźniających palenie się na ogólną toksyczność pianek poliuretanowych. Znaczna większość literatury wskazuje, że dodanie opóźniaczy ognia nie zwiększa toksyczności pianek poliuretanowych. Wynika to z dużego zakresu dostępnych środków opóźniających palenie się, które można zastosować do produkcji pianek poliuretanowych.

⁵ Gajlewicz I., Lenartowicz M., *Nowe kierunki uniepalniania tworzyw polimerowych*. Przetwórstwo Tworzyw, 3, 216-224 (2014).

⁶ <https://www.chemicalsafetyfacts.org/flame-retardants/> (dostęp na 12.12.2021)

3. CHARAKTERYSTYKA WYBRANYCH UNIEPALNIACZY

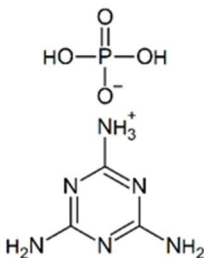
Melamina oraz jej pochodne to jeden z najczęściej stosowanych uniepalniaczy (Rys. 1), co jest spowodowane niskim kosztem. Jest to proszek, podobnie jak jej sól – izocyjanuran melaminy (Rys. 2) również ma taką samą formę. Melamina jest skutecznym środkiem zmniejszającym palność, jednak stosowanie jej z innymi antypirenami jest konieczne aby poprawić właściwości mechaniczne a nawet właściwości pożarowe otrzymywanych produktów. Zawiera ona aż 67% azotu i podczas rozkładu w wysokiej temperaturze wydzielają się z melaminy amoniak tworzący odporne termicznie kondensaty.



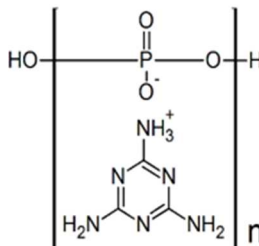
Rysunek.1. Melamina

Rysunek.2. Izocyjanuran melaminy

Fosforan melaminy (Rys.3) powstaje w reakcji kwasu fosforowego i melaminy. Azot związany w pierścieniu melaminy uwalnia się z fosforanów melaminy przechodząc do fazy gazowej w wysokiej temperaturze przy czym pochłania ciepło z otoczenia.⁷ Poli(fosforan melaminy) to polimeryczna forma fosforanu melaminy (Rys. 4).



Rysunek.3. Fosforan melaminy



Rysunek.4. Poli(fosforan melaminy)

⁷ Cichy B. Fosforany melaminy jako przyjazne ekologicznie, bezhalogenowe reterdanty palenia materiałów polimerowych. Chemik 2013, 214-219

Roflam F5 to niepalniacz fosforowy, bezhalogenowy - w związku z czym podczas spalania w fazie gazowej nie wydzielają się toksyczne gazy takie jak np. HBr czy HCl. Podczas spalania działa w fazie stałej – tworzy on na powierzchni materiału zwęgloną warstwę, która hamuje rozprzestrzenianie się płomienia.⁸

Roflam P to bardzo skuteczny niepalniacz ze względu na obecność w swojej budowie chloru oraz fosforu. W temp. pokojowej jest cieczą. Bardzo dobrze działa z niepalniaczami proszkowymi takimi jak np. melamina dając pianki poliuretanowe o bardzo wysokiej odporności na ogień.⁹

Roflam B7L jest bezhalogenowym niepalniaczem fosforowym. W temperaturze pokojowej jest cieczą. Jest on całkowicie bezpieczny dla zdrowia człowieka i dla środowiska. Jako, że jest bezhalogenowy to nie wydziela toksycznych gazów podczas spalania takich jak np. chlorowodór.¹⁰

Grafit ekspandowany wytwarzany jest przez eksfoliację kryształów grafitu interkalowanego atomami siarki albo fluoru. Jego działanie jako antypirenu opiera się na wytworzeniu zwęglonej warstwy która izoluje materiał od płomienia- dzieje się tak dzięki formacji niewielkich luk powietrznych pomiędzy warstwami Grafitu.¹¹

Exolit to antypiren zawierający fosfor. Jako reaktywny środek zmniejszający palność może wchodzić w reakcje chemiczną z matrycą PU, zapobiegając niepożądaną migracji z materiału. Jako niepalniacz bezhalogenowy jest korzystny dla zdrowia i środowiska.¹²

4. BEZPIECZEŃSTWO POŻAROWE

W zastosowaniach związanych z bezpieczeństwem pożarowym, jednym z ważnych czynników jest określenie, czy płomień zainicjowany w jednym punkcie ograniczy się do lokalnego, chwilowego spalania, czy też rozwinie się w duży pożar. Inną kwestią jest określenie, jak szybko płomień rozprzestrzenia się, aby objąć więcej płonących powierzchni. Szybkość rozprzestrzeniania się może być również oszacowana poprzez pomiar czasu potrzebnego na przemieszczenie się frontu płomienia z jednego punktu do drugiego na powierzchni palącego się materiału. Pianki zawierające odpowiednio dobrane środki niepalniające wykazują właściwości samogasnące, czyli nie będą powodować dalszego rozprzestrzeniania się pożaru. Pianka samoistnie zacznie się wygaszać po usunięciu źródła ognia. Na jej powierzchni zostanie tylko ciemny zwęglony ślad.¹³

⁸ <https://www.products.pcc.eu/pl/id/1077/roflam-f5/> (dostęp: 22.04.2022)

⁹ <https://www.products.pcc.eu/pl/id/8908/roflam-p/> (dostęp: 10.11.2021)

¹⁰ <https://www.products.pcc.eu/pl/id/8432/roflam-b7l/> (dostęp: 10.11.2021)

¹¹ Bhagat V., Polyurethane foam association, 2001, 5, 1-5

¹² <https://www.clariant.com/es/Business-Units/Additives/Flame-Retardants/Product-Line-Overview/Exolit-OP> (dostęp: 22.04.2022)

¹³ <https://jetpur.pl/czy-piana-pur-jest-palna-co-z-ta-palnoscia-odpowiadamy/> (dostęp na dzień 10.11.2021)

5. PODSUMOWANIE

Środki zmniejszające palność to bardzo zróżnicowana grupa związków, które są dodawane do materiałów w celu zmniejszenia ich palności. Na rynku jest dostępnych dużo różnych rodzajów tych substancji, a dodatkowo ich mieszanki w różnych proporcjach dają inne, zwykle lepsze wyniki niż zastosowane pojedynczo.

Właściwy dobór środków zmniejszających palność nie jest łatwy ze względu na dużą ilość materiałów polimerowych różniących się budową czy składem. Każdy z uniepalniaczy może inaczej zachowywać się w danym związku, który chcemy uniepalniać. Warto więc badać te substancje pod kątem właściwości pożarowych aby otrzymywać coraz to lepsze wyniki.

Dodatkowo (oprócz właściwości pożarowych) należy badać pozostałe parametry otrzymanych produktów. Każdy uniepalniacz wprowadzony do pianki poliuretanowej wywołuje zmianę wielu właściwości np. mechanicznych czy fizycznych. Znaczna większość literatury wskazuje, że dodanie opóźniaczy ognia nie zwiększa toksyczności pianek poliuretanowych. Wynika to z dużego zakresu dostępnych środków opóźniających palenie się, które można zastosować do produkcji pianek poliuretanowych.

LITERATURA

1. Janowska G., Przygocki W., Włochowicz A., *Palność polimerów i materiałów polimerowych*, WNT, Warszawa 2007, 25-26, 187, 199-204.
2. Hull T. R., Stec A. A., *Polymers and fire*, Fire Retardancy of Polymers: New Strategies and Mechanisms, Royal Society of Chemistry, 2009, 1-14.
3. Singh H., Jain A.K., *Ignition, combustion, toxicity, and fire retardancy of polyurethane foams: A comprehensive review*. Journal of Applied Polymer Science articles, 10, 2008, 1-9.
4. Gajlewicz I., Lenartowicz M., *Nowe kierunki uniepalniania tworzyw polimerowych*. Przetwórstwo Tworzyw, 3, 216-224 (2014).
5. Cichy B. Fosforany melaminy jako przyjazne ekologicznie, bezhalogenowe reterdanty palenia materiałów polimerowych. Chemik 2013, 214-219.
6. Bhagat V., Polyurethane foam association, 2001, 5, 1-5.

ŹRÓDŁA INTERNETOWE

7. <https://www.element.com/materials-testing-services/flammability-test-methods> (dostęp na dzień 10.11.2021)
8. <https://www.chemicalsafetyfacts.org/flame-retardants/> (dostęp: 12.12.2021)
9. <https://www.products.pcc.eu/pl/id/8908/roflam-p/> (dostęp: 10.11.2021)
10. <https://www.products.pcc.eu/pl/id/8432/roflam-b7l/> (dostęp: 10.11.2021)
11. <https://www.products.pcc.eu/pl/id/1077/roflam-f5/> (dostęp: 22.04.2022)

-
12. <https://www.clariant.com/es/Business-Units/Additives/Flame-Retardants/Product-Line-Overview/Exolit-OP> (dostęp: 22.04.2022)
 13. <https://jetpur.pl/czy-piana-pur-jest-palna-co-z-ta-palnoscia-odpowiadamy/> (dostęp: 10.11.2021)

Klaudia STACHNIK

dr inż. Dorota GŁOWACZ–CZERWONKA prof. PRz
opiekun naukowy

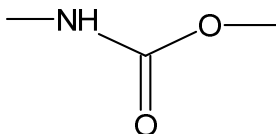
UNIEPALNIONE, SZTYWNE PIANKI POLIURETANOWE NA BAZIE ROKOPOLU RF 151V. CZ. 2. BADANIE PALNOŚCI KOMPOZYCJI

W dzisiejszych czasach, przemysł tworzyw sztucznych odgrywa istotną rolę. Dlatego też poszukuje się takich tworzyw, które zapewniają bezpieczeństwo w ich eksploatacji. Obiektem badań w tym artykule są sztywne pianki poliuretanowe (z dodatkiem antypirenów addytywnych), które zbadano pod kątem palności. Wyznaczono indeks tlenowy i przeprowadzono test poziomy. Dzięki temu zaobserwowano, że dodatek Exolitu, jako antypirenu addytywnego, przyczyniła się do znacznego polepszenia właściwości ognioodpornych otrzymanych kompozycji poliuretanowych.

Słowa kluczowe: sztywne pianki poliuretanowe, PUR, palność, antypireny addytywne,

WPROWADZENIE

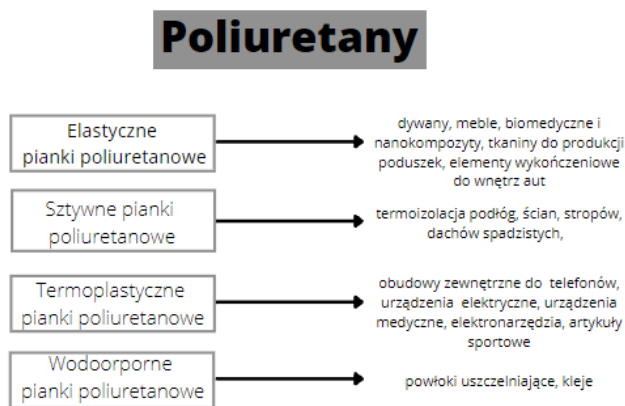
Poliuretany są to specyficzne materiały należące do grupy polimerów liniowych lub usieciowanych, które wykazują odmienność od innych tworzyw sztucznych. Charakterystyczną ich cechą jest powtarzające się w ich budowie ugrupowanie uretanowe przedstawione na rys. 1.



Rysunek.1. Ugrupowanie uretanowe.

Poliuretany występują w głównej mierze pod postacią pianek poliuretanowych. Obecnie stosowane są jako dodatki do wielu wyrobów przemysłowych z różnych dziedzin. Wskazuje to na ich wszechstronne zastosowanie, a to natomiast dowodzi w jak ogromnej skali zostały one rozpowszechnione na całym świecie.

Poniższy schemat przedstawia główne rodzaje materiałów poliuretanowych, a także przykłady ich zastosowania^{1,2}.



Rysunek.2. Typy pianek poliuretanowych i ich zastosowanie².

Początkowo zamysł wynalezienia PU miał służyć zastąpieniu gumy, która była wykorzystywana podczas II Wojny Światowej. Nikt nie spodziewał się, że ten materiał zyska taką wszechstronność. Jak się później okazało, wiele materiałów, które miały swoje wady lub były trudno dostępne, bądź po prostu drogie, można było zastąpić tworzywem poliuretanowym. Zapoczątkowało to stosowanie go jako powłok, których dodatek miał za zadanie poprawiać właściwości końcowego produktu, którym m.in. jest papier, odzież. Powłoki te wykorzystywano również do zabezpieczania powierzchni drewna, metalu, a nawet do stworzenia wykończenia o wysokim połysku przeznaczonym dla samolotów¹.

Za sprawą Otto Bayer'a pianki poliuretanowe stały się zdobyczą technologiczną dzisiejszych czasów. Obecny przemysł jest na tyle rozwinięty, a zapotrzebowanie na różnego rodzaju produkty tak wielkie, że materiały poliuretanowe, w tym pianki, znalazły zastosowanie począwszy od budownictwa, a kończąc

¹ Zafar F., Sharmin E., *Polyurethane*, IntechOpen, Croatia, 2012

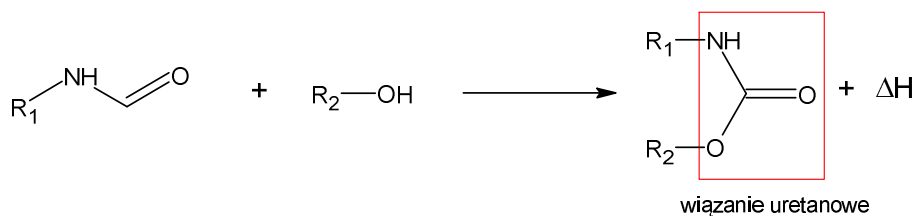
² Akindoyo J. O., Beg, Suriati Ghazali M. D. H., Islam M. R., Nitthiyah Jeyaratnam, Yuvaraj A. R., *Polyurethane types, synthesis and applications – a review*, RSC Advances, 115, 2016, str.

na medycynie, co stało się ich ogromnym atutem. Jest to ogromny wachlarz możliwości w ich wykorzystaniu³.

Pianki poliuretanowe to materiały charakteryzujące się porowatą strukturą, którą zyskują dzięki poroforom. Zadaniem substancji porotwórczych jest wytworzenie w reakcji gazu, najczęściej CO₂, który następnie tworzy pęcherzyki w całej objętości pianki⁴.

Pianki powstają w reakcji poliaddycji lub polimeryzacji stopniowej, w której grupy hydroksylowe, pochodzące od polioli, reagują z grupami izocyjanianowymi. Powstały związek ma w swojej budowie wiązania uretanowe, kluczowe w całym tym procesie.

Reakcja zachodzi z wydzieleniem się pewnych ilości ciepła, zatem ważna jest kontrola temperatury w trakcie procesu. Wodór pochodzący od grupy hydroksylowej przyłącza się do azotu z grupy izocyjanianowej, a z węglem łączy się pozostały tlen razem z resztą R₂ pochodzącą od polioli (Rys.3).⁵



Rysunek.3. Schemat powstania wiązania uretanowego⁶.

Obiektem badań w artykule są sztywne pianki poliuretanowe. Najważniejszym ich zastosowaniem, dzięki któremu zyskały globalne zainteresowanie, jest termoizolacja. Przyczyną tych unikalnych właściwości jest ich struktura komórkowa⁶. Są one szeroko stosowane w przemyśle budowlanym, chłodnictwie oraz w przemyśle rurowym. Jednak ze względu na podatność na ogień, wiele chemików szuka jak najskuteczniejszej i najtańszej metody modyfikacji pianek⁷.

Sztywne pianki poliuretanowe, będące obiektem badań, znalazły szerokie zastosowanie w przemyśle budowlanym jako materiał izolacyjny, można się ich również doszukać w chłodnictwie⁶. Poniżej na Rys. 4 przedstawiono ciekawe

³ Demharter A., Polyurethane rigid foam, a proven thermal insulating material for applications between +130°C and -196°C, *Cryogenics*, vol. 38, 1998, 113

⁴ Szycher M., *Szycher's handbook of polyurethanes*, CRC Press LLC, New York 1999, 157-158

⁵ Gama N. V., Ferreira A., Barros-Timmons A., *Polyurethane Foams: Past, Present and Future*, *Materials* 2018, 11 (10), 1841

⁶ Günther M., Lorenzetti A., Schartel B., *Fire phenomena of rigid polyurethane foams*, *Polymers*, 2018, 10 (10), 1166

⁷ Zatorski W., Brzozowski Z. K., Kolbrecki A., *New development in chemical modification of fire-safe rigid polyurethane foams*, *Polymer Degradation and Stability*, 2008, 93 (11), 2071

wykorzystanie pianek zaproponowane przez firmę JAG PPH Sp. z o.o. Firma przedstawia się jako producent pianek o najmniejszym współczynniku przewodzenia ciepła i wykazujących klasę palności F1313.



Rysunek.4. Zastosowanie pianek poliuretanowych według firmy JAG PPH Sp. z o.o.⁸

Kolejnym kierunkiem zastosowania jest użycie sztywnych pianek poliuretanowych w postaci natryskowej. Powstają one poprzez połączenie się składnika A, którym jest mieszanina składająca się z polioli oraz środków pomocniczych oraz składnika B czyli izocyjanianu. Następnie przed nanoszeniem pianki następuje mechaniczne wymieszanie. Pianka zapewnia termoizolację poprzez wyeliminowanie tzw. mostków cieplnych⁹.

Szttywne pianki poliuretanowe wykazują właściwości termoizolacyjne, ale również mogą służyć do izolacji akustycznej. W pierwszym przypadku, komórki są zamknięte, a drugim - otwarte. Dodatkowo, jako wzmocnienie, stosuje się włókno szklane, dzięki któremu pianka zyskuje wytrzymałość przy jednoczesnym zachowaniu wagi. Materiał ten sam z siebie jest stosowany do izolacji cieplnej oraz akustycznej.

Poliuretany są polimerami, które w kontakcie z ogniem ulegają całkowitemu spalaniu. Z tego względu przeprowadzono już wiele badań, które mają na celu otrzymanie produktu odpornego termicznie. Chcąc uzyskać tę właśnie właściwość, stosuje się specjalne związki spowalniające proces palenia, a nawet zapobiegające zajęciu się materiału, nazywają się one antypirenami lub inaczej inhibitorami spalania. Degradacja wiązania uretanowego w sztywnych piankach poliuretanowych rozpoczyna się w temperaturze 200-250°C¹⁰. Z tego względu wymagane jest stosowanie środków poprawiających odporność termiczną. Działają one na zasadzie zwiększenia pojemności cieplnej danego materiału lub po-

⁸ JAG PPH Sp. z o.o., Pianka PUR/PIR-szttywna pianka poliuretanowa (Typ PUR), Dostępne online: <https://www.jag.pl/pl/oferta/szttywna-pianka-poliuretanowa.html>

⁹ Dudek P, Możliwości zastosowania pianek poliuretanowych do termoizolacji domów, Dostępne online: <https://www.ekspertbudowlany.pl/arttykul/dachy/182126.mozliwosci-zastosowania-pianek-poliuretanowych-do-termoizolacji-domow>

¹⁰ Bilbao R., Mastral J.F., Ceamanos K., M.e. Aldea, Kinetics of the thermal decomposition of polyurethane foams in nitrogen and air atmosphere, Journal of Analytical and Applied Pyrolysis, 7, 1996 str.69-82

wstaniu warstwy zabezpieczającej pod postacią gazu niepalnego lub piany. Aby osiągnąć odporność na rozprzestrzenianie się ognia, należy zredukować ilość palnych fragmentów, zmniejszyć szybkość uwalniania się palnych gazów oraz uniknąć długotrwałego spalania.

Antypireny dzieli się na addytywne i reaktywne.

Pierwsze z wymienionych, nie wchodzi w reakcję z danym polimerem, są one dodawane przed, w trakcie lub po procesie polireakcji. Natomiast antypireny reaktywne wykazują zdolność wbudowania się w strukturę związku, który należy uniepalnić^{11,12}.

Dobrze dobrany antypiren powinien charakteryzować się następującymi cechami:

- mieć niską cenę,
- nie może działać jako plastyfikator,
- musi zapewnić trwałość ognioodporności danego materiału,
- wykazywać niską lepkość,
- być odporny na hydrolizę,
- nie mieć niekorzystnego wpływu na właściwości produktu właściwego,
- nie emitować dużej ilości toksycznych gazów oraz dymów w trakcie spalania,
- być kompatybilny ze wszystkimi składnikami stosowanymi do otrzymania PU¹³.

Bardzo często stosuje się mieszaniny inhibitorów spalania, a służy to otrzymaniu materiału wysoce ognioodpornego.

1. CZĘŚĆ DOŚWIADCZALNA

Pianki poliuretanowe otrzymano z handlowego polioliu Rokopolu RF 151V (PCC Rokita, Brzeg Dolny), MDI (Borsodchem, Hungary), stosując środki zwiększające odporność na płomień: melaminę (Mel), dipentaerytrytol (DPE), izocyjanuran melaminy (ICC), fosforan melaminy (MP), poli(fosforan melaminy) tj. MPP, grafit ekspandowany (EG 096), dipentaerytrytol (DPE), exolit (Ex), Roflam F5 i B7 (RF5 i RB7), a także hydroksycynian cynku (HCC) i kwas borowy (KB).

W celu zbadania palności pianek wyznaczono indeks tlenowy otrzymanych kompozycji, a także przeprowadzono test poziomy.

¹¹ Porowski R., Wprowadzenie do analizy termicznej polimerów, Zeszyty Naukowe SGSP/Szkoła Główna, Służby Pożarniczej, 2017, 4, 67-90

¹² Gajewicz I., Lenartowicz M., Nowe kierunki uniepalniania tworzyw polimerowych, *Przetwórstwo tworzyw*, 3, 2014. 218

¹³ *Ibidem*

2. WYNIKI

Indeks tlenowy (LOI) uchodzi za podstawowe badanie, dzięki któremu możliwe jest określenie ognioodporności. Pozwala ono ustalić minimalną ilość tlenu, która jest potrzebna do podtrzymywania procesu spalania pianki. Parametr ten jest wyrażony w procentach objętościowych zawartości tlenu i azotu.

Ognioodporność klasyfikuje się następująco¹⁴: LOI < 21% - materiał palny, 21% < LOI < 28% - materiał samogasnący, 28% < LOI < 100% - materiał niepalny.

Otrzymane wyniki zaprezentowano w tabeli 1.

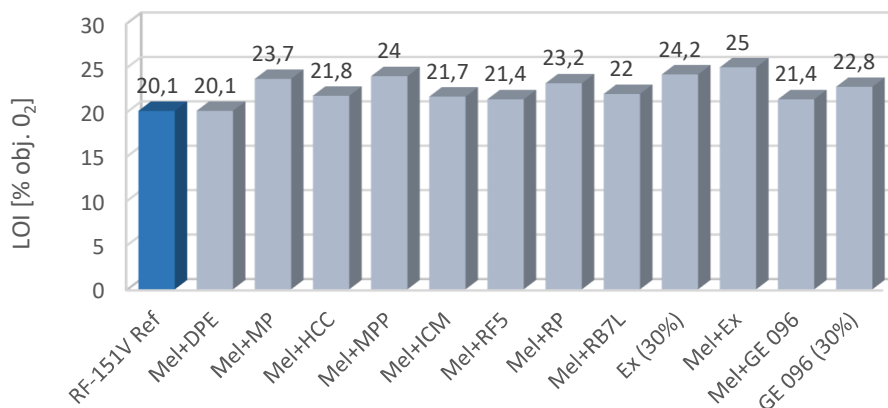
Tabela 1. Zestawienie otrzymanych wyników wartości indeksu tlenowego.

Lp.	Skrót kompozycji	LOI	Klasyfikacja palności
1.	RF-151V Ref	20,1	palna
2.	Mel+DPE	20,1	palna
3.	Mel+MP	23,7	samogasnąca
4.	Mel+HCC	21,8	samogasnąca
5.	Mel+MPP	24	samogasnąca
6.	Mel+ICM	21,7	samogasnąca
7.	Mel+RF5	21,4	samogasnąca
8.	Mel+RP	23,2	samogasnąca
9.	Mel+RB7L	22	samogasnąca
10.	Ex (30%)	24,2	samogasnąca
11.	Mel+Ex	25	samogasnąca
12.	Mel+GE 096	21,4	samogasnąca
13.	GE 096 (30%)	22,8	samogasnąca

Badaniu poddano wcześniej przedstawione kompozycje, dla większości z nich, bo 11 z 13, uzyskano wynik powyżej 21%, co klasyfikuje je jako materiały samogasnące. W przypadku pianki referencyjnej, która nie zawierała dodatku antypirenow oraz pianki z melaminą i dipentaerytrytolem zaobserwowano podatność na ogień.

Na podstawie danych zamieszczonych w tabeli, wykonano wykres 1.

¹⁴ Norwiński S., Postawa P., Ocena palności metodą indeksu tlenowego (LOI) wybranych kompozytów na osnowie polipropylenu, Częstochowa 2016



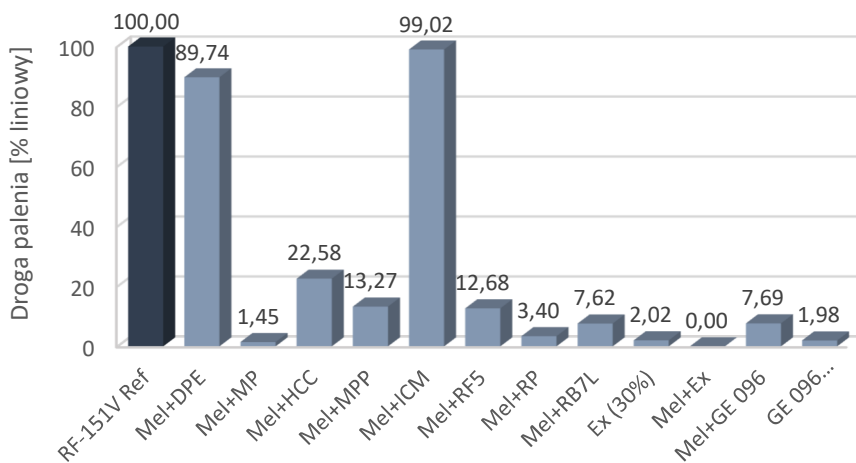
Wykres.1. Wartości LOI dla poszczególnych kompozycji.

Na wykresie, ciemniejszym kolorem zaznaczono piankę referencyjną. Największą wartość indeksu tlenowego, wynoszącą 25% obj. O₂, odnotowano dla kompozycji piankowej zawierającej Melaminę i Exolit.

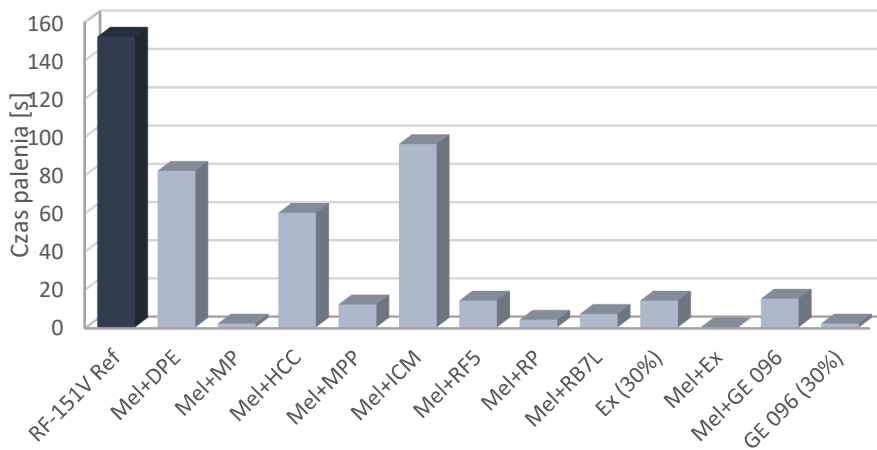
Do wykonania testu poziomego przygotowano kształtki, na których (przed poddaniem ich działaniu ognia), zaznaczano linię znajdującą się w odległości 2,5 cm od krótszej krawędzi. W przypadku przekroczenia wyznaczonej linii, piankę uznawano jako materiał palny.

Podczas badania mierzono czas od momentu zapalenia się kształtki do jej całkowitego zgaśnięcia. Dla porównania przebiegu procesu spalania wykonano wykresy drogi palenia (wykres 2), czasu palenia (wykres 3) oraz ubytku masy (wykres 4) dla poszczególnych kompozycji piankowych.

W pierwszej kolejności warto zacząć od porównania procentowej drogi przebytej przez ogień. Jak można było się spodziewać, kompozycja RF-151V Ref uległa całkowitemu spalaniu. W przypadku pozostałych pianek wprowadzenie uniepalniaczy sprawiło ograniczenie rozprzestrzeniania się ognia, czyli pianka ulegała samogaśnięciu. Wskazuje to na osiągnięcia zamierzonego zamiaru, którym był osiągnięcie kompozycji uniepalnionych. Jednak w dwóch przypadkach efekt uniepalnienia nie był zadawalający (kompozycje Mel+DPE oraz Mel+ICM). Ich drogi palenia osiągnęły odpowiednio 89,74% liniowych oraz 99,02% liniowych. Są to zbyt duże wartości, aby mogłyby być one bezpiecznie stosowane jako izolacja.

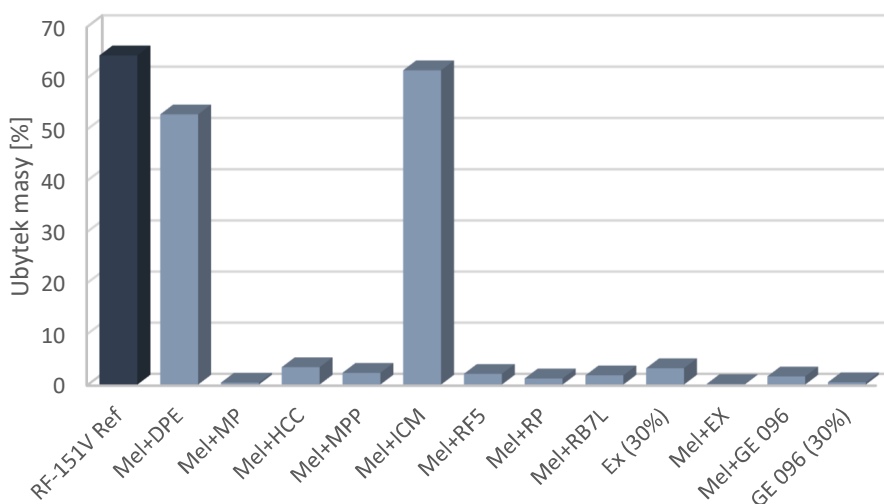


Wykres. 2. Droga palenia kształtek [% liniowy]..



Wykres. 3. Czas palenia otrzymanych kompozycji piankowych.

Czas palenia jest ściśle związany z drogą przebytą przez ogień. Im dłuższa droga, tym dłuższy czas palenia. Czas palenia jest najdłuższy w wypadku kompozycji referencyjnej (RF-151VRef.), długie palenie dotyczy także pianek: Mel+ICM (96 s), Mel+DPE (82 s) oraz Mel+HCC (60 s). Spalenie całej powierzchni próbki dla pianki referencyjnej wyniosło 152 s i jest to zarazem najdłuższy osiągnięty wynik. Drugim z kolei czasem jest 96 s, który należy do kompozycji Mel+ICM. Jak można zauważyć, w przypadku pianki Mel+HCC, proces palenia zajął 60 s, jednak porównując go do drogi przebytej przez płomień, to 22,6% liniowych jest dość dużą wartością. Spalenie 89,7% kształtki Mel+DPE zajęło 82 sekundy, a jest to jedynie 22 sekundy dłużej. Wskazuje to na fakt, iż pianka z dodatkiem HCC wykazuje właściwości samogasnące, lecz do całkowitego wygaszenia wymagane jest więcej czasu, niż w przypadku pozostałych pianek.









Wykres. 4. Ubytek masy poszczególnych próbek.

Na wykresie 4 przedstawiono ubytek masy poszczególnych próbek. Największa strata masy zauważalna jest w przypadku pianki referencyjnej, co jest oczywiste ze względu na jej całkowite spalanie. Drugą w kolejności jest kompozycja, która zawiera melaminę z izocyjanuranem melaminy, a trzecią zawierająca melaminę i dipentaerytrytol. W przypadku wymienionych pianek zauważyć można wyraźny związek z drogą palenia.

Zmianę wyglądu kształtek poddanych testowi poziomemu przedstawiono w tabeli 2.

Tabela.2. Zestawienie wyglądu kształtek przed i po teście poziomym.

Skrót kompozycji	RF-151V Ref	Exolit (30%)	GE 096 (30%)
Przed			
Po			

Kolejnymi istotnymi danymi, które zostały zanotowane są ubytki masy kształtek. Pozwalają one na oszacowanie jakich ubytków, w przypadku materiału izolacyjnego, można się spodziewać, gdy dojdzie do jego kontaktu z ogniem. Ważnym jest, aby wartość ta była jak najmniejsza. Większość pianek uzyskała zadowalające wyniki. Największy ubytek masy został zarejestrowany dla RF-151V Ref, która uległa całkowitemu spaleniu oraz dla Mel+ICM. Wartości te wynoszą odpowiednio 64,2 % i 61,4%. Drugą pianką, wykazującą znacznie większy ubytek względem pozostałych próbek, jest Mel+DPE z wynikiem 52,8%. Można było się spodziewać takiej zależności, ponieważ dla wymienionych kompozycji droga palenia okazała się największa.

3. PODSUMOWANIE

Celem artykułu było przedstawienie badań jakie przeprowadzono na otrzymanych sztywnych piankach poliuretanowych. Poszukiwano możliwie najlepszej kompozycji antypirenów addytywnych, które zapewnią ogniodporność. Zaobserwowano, że najlepszy wpływ na uodpornienie ich miał Exolit OP 935 w pojedynkę, ale i również z dodatkiem Melaminy. Pianki zawierające go uzyskały najlepsze wyniki w badaniu indeksu tlenowego, ale również w teście poziomym.

Exolit OP 935 zapewnia tak dobre właściwości ogniodporne ze względu na wysoką zawartość fosforu w swojej strukturze. Jego zaletą jest to, że nie zawiera on halogenków, co skutkuje korzystnym stosunkiem względem środowiska oraz zdrowia. Z tego też powodu chętnie stosuje się go jako dodatek do materiałów, np. sztywnych pian poliuretanowych przeznaczonych do uszczelniania powierzchni, które są narażone na kontakt z ogniem.

LITERATURA

1. Akindoyo J. O., Beg, Suriati Ghazali M. D. H., Islam M. R, Nitthiyah Jeyaratnam, Yuvaraj A. R., *Polyurethane types, synthesis and applications – a review*, RSC Advances, 115, 2016
2. Bilbao R., Mastral J.F., Ceamanos K., M.e. Aldea, *Kinetics of the thermal decomposition of polyurethane foams in nitrogen and air atmosphere*, Journal of Analytical and Applied Pyrolysis, 7, 1996
3. Demharter A., *Polyurethane rigid foam, a proven thermal insulating material for applications between +130°C and -196°C*, Cryogenics, 38, 1998, 113
4. Gajewicz I., Lenartowicz M., *Nowe kierunki uniepalniania tworzyw polimerowych*, Przetwórstwo tworzyw, 3, 2014. 218
5. Gama N. V., Ferreira A., Barros-Timmons A., *Polyurethane Foams: Past, Present and Future*, Materials 2018, 11 (10), 1841
6. Günther M., Lorenzetti A., Schartel B., *Fire phenomena of rigid polyurethane foams*, Polymers, 2018, 10 (10), 1166
7. Norwiński S., Postawa P., *Ocena palności metodą indeksu tlenowego (LOI) wybranych kompozytów na osnowie polipropylenu*, Częstochowa 2016
8. Porowski R., *Wprowadzenie do analizy termicznej polimerów*, Zeszyty Naukowe SGSP/Szkoła Główna Służby Pożarnej, 2017, 4, 67-90
9. Szycher M., *Shycher's handbook of polyurethanes*, CRC Press LLC, New York 1999, 157-158
10. Zafar F., Sharmin E., *Polyurethane*, IntechOpen, Croatia, 2012
11. Zatorski W., Brzozowska Z. K., Kolbrecki A., *New development in chemical modification of fire-safe rigid polyurethane foams*, Polymer Degradation and Stability, 2008, 93 (11), 2071

ŹRÓDŁA INTERNETOWE

12. Dudek P, *Możliwości zastosowania pianek poliuretanowych do termoizolacji domów*, Dostępne online: <https://www.ekspertbudowlany.pl/arttykul/dachy/182126,mozliwosci-zastosowania-pianek-poliuretanowych-do-termoizolacji-domow>
13. JAG HPP Sp. z o.o., *Pianka PUR/PIR-sztywna pianka poliuretanowa (Typ PUR)*, Dostępne online: <https://www.jag.pl/pl/oferta/sztywna-pianka-poliuretanowa.html>



KOŁO

NAUKOWE



OGRZEWANIE

WENTYLACJA

KLIMATYZACJA



Justyna KUNA, Agata OLEJARCZUK, Oskar RUMAK

mgr inż. Paweł KUT
opiekun naukowy

POMIARY SKUTECZNOŚCI WENTYLACJI MECHANICZNEJ WYWIEWNEJ W BUDYNKACH UŻYTECZNOŚCI PUBLICZNEJ

Artykuł opisuje znaczenie wentylacji w użytkowaniu budynków oraz opisuje sposoby wymiany powietrza. Przedstawiono w nim podział wentylacji naturalnej i mechanicznej oraz zaznaczono, dlaczego stosowanie wentylacji mechanicznej zaczęło być istotne. Wyszczególniono wady oraz zalety powyższej metody wentylacji oraz opisano sposoby wentylacji w obiektach sakralnych, w tym wentylację wywiewną. Zostały przedstawione wyniki badań, którego celem było zbadanie wilgotności powietrza w obiekcie sakralnym przy Lwowskiej 125 w Rzeszowie z wentylacją mechaniczną oraz bez niej. Badania zostały przeprowadzone czujnikiem wilgotności i temperatury, dzięki któremu otrzymane wyniki ukazały skuteczność w usuwaniu zużytego powietrza dzięki wentylacji mechanicznej wywiewnej w danym obiekcie.

Słowa kluczowe: wentylacja, wilgotność.

WPROWADZENIE

Wymagany stopień komfortu czy też jakość mikroklimatu jest obecnie najważniejszym czynnikiem warunkującym dobre samopoczucie człowieka przebywającego w pomieszczeniu zależy on nie tylko od temperatury, ale także parametrów ciepłno-wilgotnościowych: stężenia zanieczyszczeń gazowych, pyłowych, mikrobiologicznych, podwyższonego hałasu¹. W wielu budynkach funkcjonujących przez dziesiątki lat, a nawet stuleci, wentylacja naturalna bardzo dobrze spełniała swoje funkcje.

Obecnie przez większą wiedzę o kosztach ogrzewania, przepływnie ciepła, czy choćby ruchu powietrza włodarze miejsc o dużej kubaturze (np. teatry, kina, obiekty sakralne), starając się obniżyć koszty ogrzewania, tworzą układy uniemożliwiające swobodne usuwanie powietrza zużytego, które przez brak wenty-

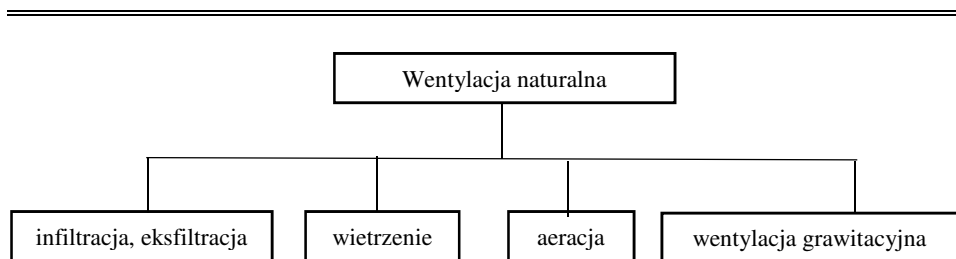
¹ A. Pelech, *Wentylacja i klimatyzacja – podstawy*, s. 11-14; 311-318; 631-633; 669-677.

lacji powodują zbyt małą wymianę powietrza oraz ograniczone usuwanie zanieczyszczeń².

Celem badania było zbadanie wilgotności powietrza w kościele parafialny oo. Pijarów pw. Św. Józefa Kalasancjusza w Rzeszowie przy ulicy Lwowskiej 125 w zależności od funkcjonowania wentylacji mechanicznej wywiewnej. Badanie pozwoliło określić skuteczność zastosowanej wentylacji mechanicznej w obiekcie.

1. SPOSOBY WYMIANY POWIETRZA

Szeroko pojęta wentylacja naturalna wystarczała, aby budynki o dużej kubaturze miały stałą wymianę powietrza. Podział wentylacji naturalnej przedstawiono na schemacie numer 1.



Schemat 1. Podział wentylacji naturalnej
Źródło: opracowanie własne

Dzięki infiltracji napływ powietrza odbywał się samoczynnie przez różnego rodzaju nieszczelności w drzwiach, oknach, a nawet przez szczeliny w przegrodach budowlanych. Wraz z infiltracją następuje eksfiltracja, czyli wypływ powietrza zanieczyszczonego z pomieszczeń. Kolejną formą wentylacji naturalnej jest przewietrzanie. Zachodziło ono dzięki różnicy ciśnień po dwóch stronach przegrody zewnętrznej. Można ją łatwo zwiększyć przez otwarcie okien, drzwi lub innych elementów przeznaczonych do tego celu. Wentylacja grawitacyjna jest następną bardzo popularną wentylacją naturalną, którą spotykamy najczęściej w naszych domach. Wymiana powietrza odbywa się w sposób samoczynny, poprzez pionowe kanały wentylacji wywiewnej, wywołana naturalnym ciągiem powietrza - powietrze dostaje się do budynku poprzez wietrzenie lub infiltrację, a wydostaje się wentylacją wywiewną. Ostatnią formą wentylacji naturalnej jest aeracja³. Stosowana jest w miejscach o dużych zyskach ciepła, najczęściej wywołane obecnością dużej liczby osób. Powietrze jest wymieniane przez

² M. Nantka, Wentylacja z elementami klimatyzacji, s. 147-167.

³ Ibidem, s. 1.

otwory wywiewno-nawiewne umieszczone w najwyższym miejscu sufitu, a dzięki różnicy ciśnień między wysokościami zachodzi samoistnie⁴.



Rysunek 1. Otwór zewnętrzny służący do usuwania powietrza przez aerację
Źródło: opracowanie własne.



Rysunek 2,3. Otwory wentylacji naturalnej w najwyższej części stropu
Źródło: opracowanie własne.

2. PROBLEMATYKA

W wyniku wentylacji naturalnej, z której wynikały duże napływy mas powietrza zimnego do pomieszczenia starano się zaizolować te pomieszczenia. W obiektach takich jak hale przemysłowe i targowe, czy miejsca kultu, rozpo-

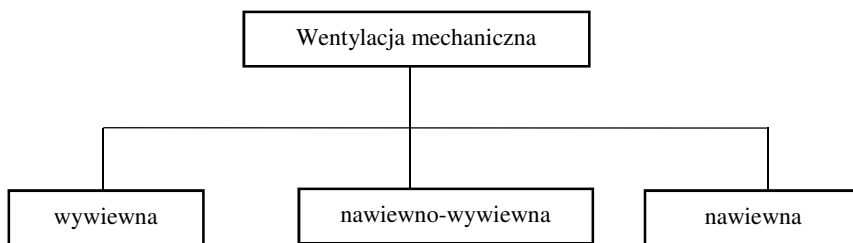
⁴ M. Nantka, *Wentylacja z elementami klimatyzacji*, op. cit., s. 147-167.

często od uszczelniania bądź wymiany okien na bardziej szczelne. Zmieniono także infrastrukturę wewnętrzną przez pomniejszenie i doszczelnienie drzwi, wymianie drzwi wahadłowych czy obrotowych na rozwierane, dobudowanie wiatrołapów. W ostatnim etapie wykonywano ocieplenia ścian budynków za pomocą materiałów izolacyjnych, a także docieplenie podłogi strychów.

Poprzez docieplenie ścian, okien i drzwi spowodowało zmniejszenie infiltracji i eksfiltracji. Dzięki zastosowaniu szczelnych wiatrołapów różnica ciśnień zmniejszyła się, przez co aeracja i wentylacja grawitacyjna zmniejszyły swoją efektywność. Zmniejszenie otworów drzwiowych i okiennych oraz zastosowanie wiatrołapów powoduje, że wietrzenie nie zachodzi tak efektywnie jak dotychczas. Z budynku, który przez wiele lat był budynkiem zdrowym, a w którym zachodziła ciągła wymiana powietrza (a któremu została polepszona izolacyjność przegród, wprowadzono nowe rozwiązania konstrukcyjne) wytworzył się układ z nadwyżką ciepła⁵. Stopień komfortu cieplnego został zachowany poprzez zmniejszenie utraty energii cieplnej, natomiast komfort – w zakresie wilgotności, stężeń zanieczyszczeń gazowych, pyłowy i mikrobiologicznych – pogorszył się⁶.

3. WENTYLACJA MECHANICZNA

Wentylację mechaniczną charakteryzuje brak wpływu warunków atmosferycznych na wymianę powietrza w danym pomieszczeniu. Za sprawą wentylatora (może to być pojedynczy wentylator, układ kilku wentylatorów) powietrze w pomieszczeniu przemieszcza się w sposób wymuszony. To inwestor określa czy chce powietrze usunąć, dostarczyć czy zarówno usunąć jak i dostarczyć. Podziału wentylacji mechanicznej przedstawiono na schemacie 2.



Schemat 2. Podział wentylacji mechanicznej

Źródło: opracowanie własne.

⁵ <http://pliki.wydawnictwo.murator.pl/ksiazeczki/ObiektySakralne/files/assets/basic-html/page66.html> (dostęp 27.05.2022r.)

⁶ <https://www.hvacr.pl/komfort-cieplny-w-kosciele-234> (dostęp 25.05.2022r.)

Wentylację mechaniczną możemy podzielić wedle wielu systemów: od kierunku ruchu powietrza w danym pomieszczeniu, różnicy ciśnień wewnątrz i na zewnątrz pomieszczenia, sposobu wymiany powietrza. Dzięki wentylacji mechanicznej najczęściej można zmieniać również parametry powietrza wentylującego.

Wentylację mechaniczną ze względu na ruch powietrza w pomieszczeniu wentylowanym dzielimy na:

- wywiewną,
- nawiewną,
- nawiewno-wywiewną.

Wentylacja wywiewna to taka, przez którą powietrze dostaje się do pomieszczenia przez różnego rodzaju nieszczelności w oknach oraz drzwiach, podobnie jak w wentylacji naturalnej. Usuwanie powietrza natomiast następuje w sposób mechaniczny. Najczęściej przez montaż wentylatorów w kanałach wentylacyjnych, które usuwają zużyte powietrze z pomieszczenia.

Kolejnym rodzajem wentylacji mechanicznej jest wentylacja nawiewna. Powietrze jest dostarczane za pomocą kanałów nawiewnych oraz wentylatorów, a usuwane w sposób naturalny⁷.

Ostatnim typem wentylacji mechanicznej jest wentylacja nawiewno-wywiewna. Powietrze jest zarówno doprowadzane jak i odprowadzane przez system wentylatorów. Powietrze świeże jest dostarczane do pomieszczenia, a powietrze zużyte jest z niego usuwane. Wentylacja nawiewno-wywiewna pozwala na utrzymanie parametrów cieplno-wilgotnościowych w pomieszczeniu.

Wentylacja mechaniczna jak wszystkie urządzenia ma swoje wady i zalety. Zalety wentylacji mechanicznej, które możemy od razu zauważyć to:

- zapewnienie stałej wentylacji powietrza we wszystkich pomieszczeniach niezależnie od warunków pogodowych,
- dzięki zastosowaniu filtrów mogą usuwać pyłki i zanieczyszczenia,
- zapewnia wymaganą ilość świeżego powietrza w pomieszczeniu,
- przy wentylacji nawiewno-wywiewnej nie ma potrzeby dodatkowego wietrzenia przez otwarcie okien, dzięki czemu ogranicza się hałas docierający spoza budynku oraz nie wytwarzają się przeciągi,
- możliwość instalacji w już zamieszkałym budynku.

Podstawowymi wadami wentylacji mechanicznej są:

- wyższe nakłady inwestycyjne (w porównaniu do wentylacji grawitacyjnej),
- konieczność doprowadzenia kanałów wentylacyjnych w budynku,
- występujące koszty eksploatacyjne (okresowa wymiana filtrów, zużycie energii elektrycznej)⁸.

⁷ A. Pełech, *Wentylacja i klimatyzacja – podstawy*, op. cit., s. 11-14; 311-318; 631-633; 669-677.

⁸ M. Nantka, *Wentylacja z elementami klimatyzacji*, op. cit., s. 147-167.

Wentylacja mechaniczna zawsze będzie lepszym wyborem, szczególnie jeśli chcemy zachować parametry powietrza, zarówno jeśli chodzi o komfort cieplny, jak również gdy będziemy chcieli zachować normy jeśli chodzi o wilgotność powietrza, czy stężenia niektórych substancji.

4. SPOSOBY WENTYLACJI W OBIEKTACH SAKRALNYCH

Każdy budynek wymaga odpowiedniej wentylacji. Wymiana zużytego powietrza na nowe jest konieczna, szczególnie dla osób przebywających w danym pomieszczeniu, a także dla dobra funkcjonowania i trwałości całego budynku oraz sprzętów znajdujących się wewnątrz⁹.

Systemy wentylacji w obiektach sakralnych różnią się w zależności od struktury budynku. Niektóre miejsca kultu zostały wybudowane stosunkowo niedawno, inne zostały wybudowane kilkadziesiąt lub kilkaset lat temu. Pierwszymi systemami wentylacyjnymi były wentylacje naturalne: wentylacja grawitacyjna, aeracja, infiltracja, eksfiltracja czy wietrzenie. Aeracja była wytwarzana przez sporych rozmiarów drzwi oraz otwory w najwyższej części stropu, przez które powietrze zostawało wyrzucane na zewnątrz, dzięki różnicy ciśnień. Infiltracja i eksfiltracja zachodziła w nieszczelnościach w drzwiach, oknach i przegrodach¹⁰. Wentylacja grawitacyjna odbywała się przez różnicę ciśnień między poziomem posadzki, a najwyższym miejscem budynku, często poprzez kanały wentylacji grawitacyjnej wbudowane w przegrody budowlane. Wietrzenie zachodziło przez otwieranie okien/drzwi, a także w budynkach budowanych jako fortyfikacje sakralno-obronne, zachodziła nieustannie przez otwory w przegrodach, które chociaż miały funkcje obronne, dla strzelających żołnierzy, wietrzyły pomieszczenie.

Systemy wentylacji w kościołach mogą być połączone ze sobą, chociaż nie muszą być ze sobą sprzężone. System wentylacji mechanicznej nawiewnej wtłacza świeże powietrze do pomieszczenia, wypychając zużyte powietrze wentylacją naturalną¹¹. Jeśli jest ono połączone z piecem, może być używane jako system ogrzewania, lub jeśli jest połączony z klimatyzatorem (ogrzewaniem w zimie i chłodzeniem w lecie). System wentylacji wywiewnej usuwa powietrze zużyte z pomieszczenia. Może być on podłączony zarówno w ścianie nieco ponad posadzką, jak i w kratkach na ścianach przy stropie, czy w samym stropie. Wentylator przetłacza powietrze zużyte i wyrzuca je na zewnątrz budynku. System wentylacji nawiewno-wywiewnej zarówno wtłacza powietrze przez nawiewniki jak i usuwa je poprzez wywiewniki.

Najczęstszym ogrzewaniem w kościołach było ogrzewanie powietrzem przez otwory w podłodze. Chociaż było częstym i, na ówczesne czasy, najeko-

⁹ <https://www.hvacr.pl/komfort-cieplny-w-kościele-234>, *op. cit.*, (dostęp 25.05.2022)

¹⁰ <https://inzynierbudownictwa.pl/drzwi-co-nalezy-wiedziec> (dostęp 28.05.2022)

¹¹ <https://www.wentylacyjny.pl/2545-49-234-nowe-warunki-techniczne-2021--wymagania-dla-wentylacji.html> (dostęp 28.05.2022).

nomiczniejszym rozwiązaniem, nie w każdym jednak kościele zdawały dobrze swoje zadanie. Obecnie uważa się za najlepsze ogrzewanie podłogowe, a kościołach w których nie można go zastosować zaleca się elektryczne maty grzewcze.

5. BADANIE WILGOTNOŚCI POWIETRZA Z WENTYLACJĄ WYWIEWNĄ

Najprostszym i najtańszym rozwiązaniem staje się mechaniczna wentylacja wywiewna. Zanieczyszczone powietrze usuwane jest przez otwory wywiewne (najczęściej adaptuje się do tego stare otwory służące do aeracji czy wentylacji grawitacyjnej)¹².



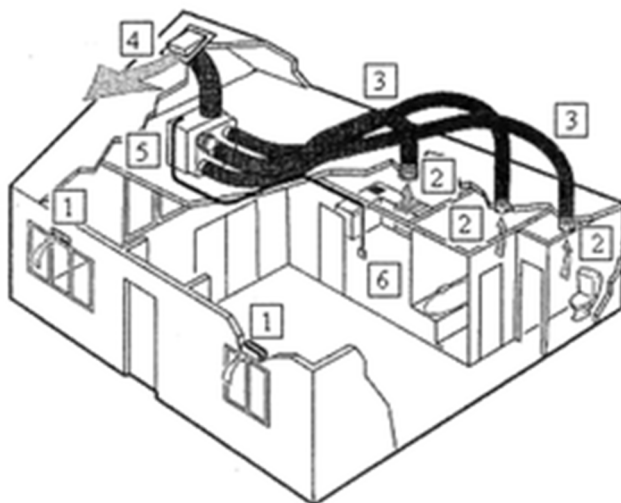
Rysunek 4. Wyrzutnia
Źródło: opracowanie własne.

Za pomocą sieci przewodów powietrze jest prowadzone do wentylatora wywiewanego, który przetłacza całe powietrze do wyrzutni poza budynek. Po włączeniu wentylacji wywiewnej powietrze zostaje zassane przez co tworzy się podciśnienie, które zwiększa objętość powietrza infiltrowanego w otworach nawiewnych¹³. ‘Dzięki obecnej technice możemy regulować objętość powietrza wywiewanego, daje to nam możliwość usuwania zużytego powietrza w zależności od warunków panujących w pomieszczeniu (takich jak ilość osób, ciśnienie, objętość tego powietrza itd.). Wentylacja mechaniczna wywiewna charakteryzuje się wytworzeniem we wnętrzu budynku podciśnień o wartościach eliminują-

¹² A. Pelech, *Wentylacja i klimatyzacja – podstawy*, op. cit., s. 11-14; 311-318; 631-633; 669-677.

¹³ M. Nantka, *Wentylacja z elementami klimatyzacji*, op. cit., s. 147-167.

cych zakłócające oddziaływanie na wymianę powietrza zmiennych warunków klimatycznych. Możemy wtedy utrzymywać w miarę możliwości wentylatora komfort ciepły i odpowiednią jakość powietrza¹⁴.



Rysunek 5. Schemat instalacji wentylacji mechanicznej wywiewnej w budynku jednorodzinym: 1- nawiewniki, 2- kratki wywiewne, 3- przewody wentylacyjne, 4- wyrzutnia powietrza, 5- wentylator wywiewny, 6- regulacja

Źródło: Nantka M., *Wentylacja z elementami klimatyzacji*, Wydawnictwo Politechniki Śląskiej, 2015 – str. 161.

Badania wilgotności powietrza zostały przeprowadzone w dniach 29.05-05.06 w kościele parafialnym oo. Pijarów pw. św. Józefa Kalasancjusza w Rzeszowie zlokalizowanego przy ulicy Lwowskiej 125. Kościół ten może pomieścić około 425 osób w części głównej mającej około 300m² powierzchni oraz około 225 osób na chórze i balkonie. Budynek został zbudowany w latach 1986-1992, następnie zajęto się pracami wykończeniowymi, które trwały do 1999r. Na przestrzeni lat zmieniały się rozwiązania techniczne samego obiektu. Zastosowano system ogrzewania powietrzem przez wyrzutnie powietrza w podłodze, na strychu zamontowano wentylator wywiewny, a kanały wentylacyjne podłączono do otworów w suficie służących wcześniej do wentylacji naturalnej, przebudowano wiatrolapy poprzez zamurowanie otworu z drzwiami wahadłowymi, a nowe drzwi ustawiono na innej ścianie. Niektóre okna zostały całkowicie wymienione

¹⁴ <https://www.izbudujemy.pl/artykuly/95/Systemy-wentylacyjne-8211-rodzaje-i-wymagania-w-swietelnowych-przepisow> (dostęp 26.05.2022)

a inne doszklone. Strop został zaizolowany pianką. Wszystkie działania miały na celu poprawę komfortu cieplnego gromadzących się ludzi.



Rysunek 6. Centrala wentylacji wywiewnej.



Rysunek 7. Przewody wentylacji wywiewnej
Źródło: opracowanie własne.



Rysunek 8. Podłączenie przewodów wentylacji mechanicznej w otworach w najwyższej części stropu
Źródło: opracowanie własne.

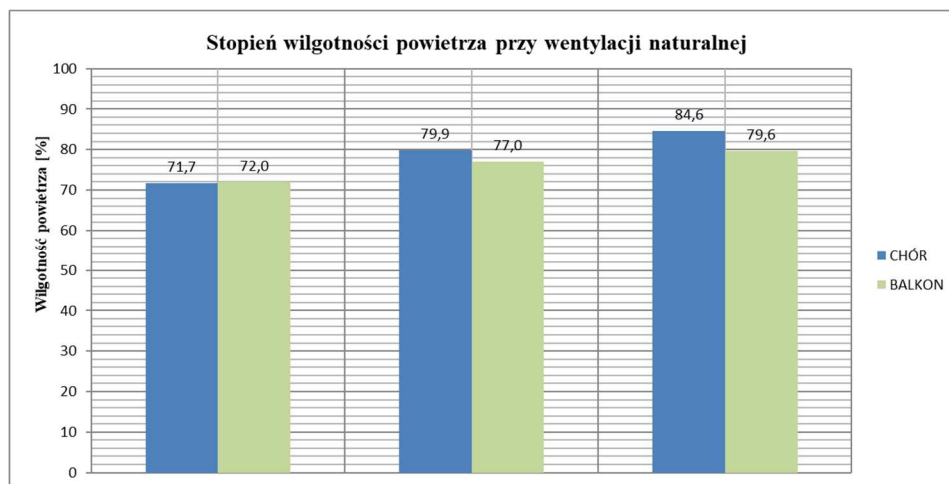
Wilgotność i temperaturę mierzono czujnikiem firmy Flus, modelem ET-965 z funkcją pomiaru temperatury i wilgotności. Pomiary wykonano bez obecności ludzi w budynku, następnie przy coraz większej liczbie osób w obiekcie w odstępach czasowych przekraczających godzinę, bez zastosowania wentylacji mechanicznej wywiewnej. Po ponad 3 godzinach włączono wentylację wywiewną, a pomiary wykonywano co około 45-75 minut. Wyniki zestawiono w tabeli poniżej.

Tabela 1. Zestawienie wyników badań

	Temperatura [°C]	Wilgotność [%]	Temperatura [°C]	Wilgotność [%]
	Chór		Balkon	
Brak wentylacji mechanicznej	18,9	71,7	19,0	72,0
	21,1	79,9	22,6	77,0
	22,4	84,6	22,6	79,6
Wentylacja wywiewna	23,1	81,7	23,6	78,3
	23,3	78,3	23,3	77,0
	23,1	73,7	23,4	74,7
	23,3	73,6	23,5	74,4

Źródło: opracowanie własne.

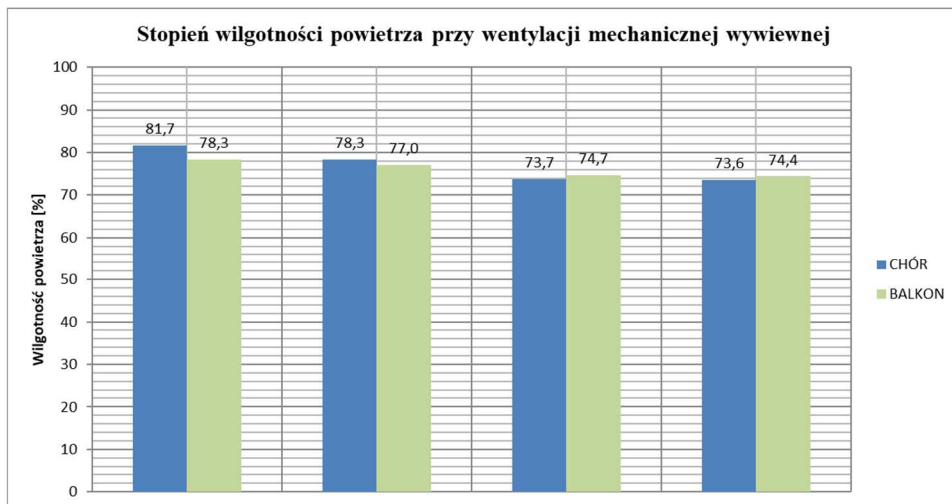
Z powyższego zestawienia możemy zauważyć obniżenie się wilgotności powietrza na chórze o 10%, a w okolicach balkonu o 5,2%. Człowiek wydychając powietrze nie tylko je ogrzewa, ale także nawilża. Im więcej osób w pomieszczeniu tym większe nie tylko stężenie dwutlenku węgla, ale także wzrost wilgotności. Poniżej przedstawiono wykres wilgotności przy wyłączonej wentylacji mechanicznej.



Rysunek 9. Stopień wilgotności powietrza przy wentylacji naturalnej
Źródło: opracowanie własne.

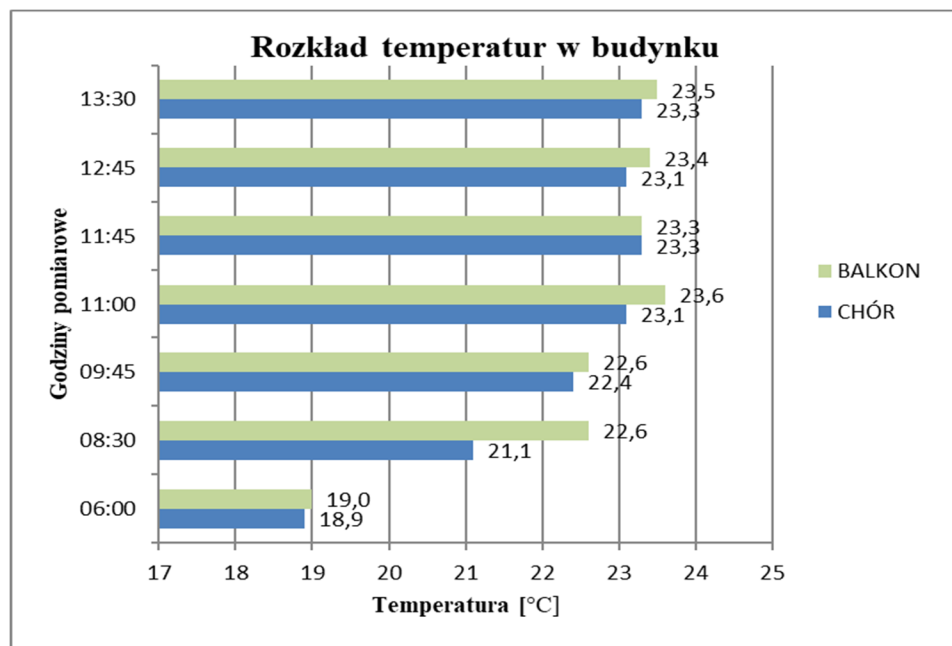
Optymalnym stopniem wilgotności w powietrzu, przy którym czujemy się dobrze, jest przedział 40-60%. Jak widzimy z powyższych badań zakres ten został przekroczony. Najwyższy wynik wyniósł na chórze 84,6 % przy temperaturze 22,4°C, a na poziomie balkonu 79,6% przy temperaturze 22,6 °C. Po włączeniu wentylacji mechanicznej już krótkim czasie widzimy rezultaty w postaci obniżenia zawartości wilgoci w powietrzu. Wyniki przedstawiono na rysunku 10.

Zużyte powietrze zostało lepiej usunięte na poziomie chóru, niż na poziomie balkonu. Jest to spowodowane układem architektonicznych budynku. Wymiana powietrza jest najbardziej niekorzystna w okolicach balkonu, ponieważ z trzech stron jest on ograniczony ścianami, z czego jedną z ścian przecinają schody prowadzące od wiatrołapów przez balkon na chór znajdujący się nad balkonem. Rozkład temperatur podczas przeprowadzenia badania przedstawiono na rysunku 11.



Rysunek 10. Stopień wilgotności powietrza przy wentylacji mechanicznej wywiewnej

Źródło: opracowanie własne.



Rysunek 11. Rozkład temperatur w budynku

Źródło: opracowanie własne.

Dzięki wytworzeniu ciągu powietrza przez wentylację mechaniczną wywiewną zużyte powietrze mogło zostać odprowadzone poza budynek, a mogło napływać świeże. Jakość powietrza została poprawiona i zbliżyła się do optymalnej zawartości wilgoci w pomieszczeniu. Dzięki większej ilości bardziej suchego powietrza nie tylko zwiększył się komfort przebywających osób wewnątrz, ale także obniżyło się ryzyko osiadania wilgoci na ścianach zimniejszych, których temperatura byłaby zbliżona do punktu rosy. Wobec powyższego nie tylko zyskaliśmy lepszy komfort jakościowy powietrza, ale także częściowo został zniwelowany problem osiadania wilgoci – nadmierne zyski ciepła zostały usunięte.

6. PODSUMOWANIE

W budynkach, które zostały zmodernizowane wentylacja naturalna przestała być wystarczająca i konieczne jest szukanie sposobów na poprawienie cyrkulacji powietrza w pomieszczeniach. Podczas modernizacji budynków należy brać pod uwagę jednocześnie stopień komfortu, jakość powietrza oraz komfort cieplny, aby wentylacja spełniała swoje funkcje należycie. Podane parametry powinny być utrzymane w wymaganych normach tak, aby komfort użytkownika budynków był jak największy. Dobrą metodą jest stosowanie mechanicznej wentylacji wywiewnej, która przy odpowiednim ustawieniu urządzeń zapewnia komfort oddychania podczas przebywania w danym pomieszczeniu oraz usuwa powietrze zużyte z budynku, a przy tym nie powiększa znacząco jego strat cieplnych. Wentylacja wywiewna jest dobrym rozwiązaniem, ponieważ niweluje nadmierne zyski ciepła. Podczas projektowania nowych budynków uwzględnienie wentylacji wywiewnej zapewnia poprawny przepływ strumienia powietrza. Wentylacja mechaniczna wywiewna usuwa nadmierne zyski ciepła z pomieszczeń, co pozwala zwiększyć komfort jakości powietrza w obiekcie. Jak wykazały badania, usuwa wilgoć, która może osiąść na ścianach i doprowadzić do dobrych warunków dla rozwoju grzybów i pleśni, wentylacja wywiewna sprawdza się w polepszeniu jakości powietrza w pomieszczeniach. Usuwa zużyte powietrze, wyrzucając je poza budynek, a przez ciąg powietrza, świeże powietrze może łatwiej dostać się do środka pomieszczenia.

Wentylacja wywiewna sprawdza się nie tylko w małych pomieszczeniach, ale i w budynkach wielko kubaturowych. Jest skuteczna w usuwaniu zużytego powietrza, wilgoci i nadmiernych zysków ciepła. Zastosowanie wentylacji mechanicznej wywiewnej jest zatem w wielu miejscach pożądane, a w niektórych wymagane.

LITERATURA

1. Pełech A., *Wentylacja i klimatyzacja – podstawy*, str. 11-14; 311-318; 631-633; 669-677.
2. Nantka M., *Wentylacja z elementami klimatyzacji*, str.: 147-167.

ŹRÓDŁA INTERNETOWE

3. <https://inzynierbudownictwa.pl/drzwi-co-nalezy-wiedziec> (dostęp 28.05.2022r.).
4. <http://pliki.wydawnictwo.murator.pl/ksiazeczki/ObiektySakralne/files/assets/basic-html/page66.html> (dostęp 27.05.2022r.)
5. <https://www.hvacr.pl/komfort-cieplny-w-kosciele-234> (dostęp 25.05.2022r.)
6. <https://www.izbudujemy.pl/artykuly/95/Systemy-wentylacyjne-8211-rodzaje-i-wymagania-w-swietle-nowych-przepisow> (dostęp 26.05.2022r.)
7. <https://www.wentylacyjny.pl/2545-49-234-nowe-warunki-techniczne-2021-wymagania-dla-wentylacji.html> (dostęp 28.05.2022r.)



KOŁO

NAUKOWE

○ INFORMATYKÓW

„KOD”



Szymon RĄPAŁA

dr inż. Bartosz TRYBUS
opiekun Koła Naukowego

SPOSOBY AUTORYZACJI DOSTĘPU DO DANYCH W APLIKACJACH WYKORZYSTUJĄCYCH REST API

Artykuł opisuje najpopularniejsze aktualnie sposoby autoryzacji dostępu do danych przez użytkowników zewnętrznych w aplikacjach wykorzystujących REST API. W artykule znajduje się opis czym jest REST API, jak działa oraz według jakich wytycznych powinno się je tworzyć. Po wprowadzeniu do działania REST API, przedstawione zostaną metody autoryzacji dostępu do danych. Na potrzeby opisu sposobów dostępu do danych wybrane zostały cztery różne metody autoryzacji. W artykule wraz z opisem każdego z czterech wybranych sposobów autoryzacji, zostanie zaprezentowany przykład ich użycia. Wybrane do przedstawienia opisy autoryzacji to podstawowa autoryzacja HTTP, klucze API, autoryzacja i uwierzytelnianie poprzez token z wykorzystaniem standardu OAuth 2.0 i OpenID Connect.

Słowa kluczowe: API, backend, token, autoryzacja, uwierzytelnianie.

WPROWADZENIE

API jest to zbiór reguł ściśle opisujący, w jaki sposób programy lub podprogramy komunikują się ze sobą. Oznacza to specyfikację wytycznych określających jako powinna wyglądać interakcja między komponentami programowymi. Specyfikacja REST została zdefiniowana pierwszy raz w 2000 roku przez informatyka dr. Roya Fieldinga w jego pracy doktorskiej. REST to zapewnienie programistom elastyczności i swobody. Połączenie REST oraz API oznacz interfejs API zgodny z zasadami projektowania REST. Wraz z rosnącą popularnością projektowania aplikacji z wykorzystaniem REST API oraz z rosnącą liczbą darmowych serwisów oferujących dane statystyczne, pojawił się problem z autoryzacją i uwierzytelnianiem dostępu do danych przechowywanych na serwerach, które są udostępniane aplikacjom zewnętrznym poprzez właśnie REST API. Przez wiele lat zostało opracowane wiele metod autoryzacji i uwierzytelniania dostępu, a cztery z nich wybrane zostaną opisane w kolejnych punktach artykułu wraz z opisem zasad projektowania REST API.

W tym miejscu warto również wyjaśnić różnicę pomiędzy uwierzytelnianiem, a autoryzacją ponieważ te dwa pojęcia będą pojawiać się w artykule dosyć często, a nie są one ze sobą tożsame. Uwierzytelnianiem nazywamy proces polegający na potwierdzeniu tożsamości, czyli kim jest osoba przesyłająca nam dane, w aplikacjach z reguły odbywa się to na podstawie nazwy użytkownika i hasła. Natomiast autoryzacja jest to proces nadawania uprawnień do zasobu, czyli co dana uwierzytelniona osoba czy aplikacja może zrobić i do jakich danych ma dostęp.

1. ZASADY PROJEKTOWANIA REST

Interfejsy oparte o specyfikację REST muszą spełniać tylko jedno wymaganie, którym jest trzymanie się sześciu zasad projektowania, znanych też jako ograniczenia architektoniczne. Pierwszą z zasad jest jednolity interfejs, wszystkie wysyłane żądania do API dotyczące tego samego zasobu powinny wyglądać tak samo, bez względu na to skąd pochodzą. Dodatkowo interfejs REST API powinien zapewniać przypisywanie identyfikatora URI do jednego elementu danych. Zasoby przedstawiane w takim interfejsie nie powinny być zbyt duże, ale jednocześnie powinny zawierać wszystkie informacje, których żądający może potrzebować.

Kolejną zasadą jest rozdział klienta i serwera, oznacza to iż podczas projektowania takiego interfejsu aplikacje klienta i serwera muszą być całkowicie od siebie niezależne. Żadna ze stron kontaktujących się ze sobą nie może modyfikować drugiej. Aplikacja klienta powinna dysponować identyfikatorem żądanego zasobu, natomiast aplikacja serwera jedynie przekazywać żądane dane za pośrednictwem protokołu HTTP, bez modyfikacji aplikacji klienta.

Trzecia zasada opisana w REST to bezstanowość. Interfejsy bezstanowe to takie, które w każdym żądaniu otrzymują wszystkie informacje niezbędne do jego przetworzenia. Co za tym idzie aplikacje serwera nie mogą przechowywać żadnych danych związanych z żądaniem aplikacji klienta.

Czwarta z kolei zasada projektowania zgodnego z REST celem zwiększenia wydajności po stronie klienta i zwiększenia skalowalności po stronie serwera wymaga udostępnienia możliwości buforowania zasobów. Możliwość buforowania powinna być dostępna po stronie klienta lub serwera.

Następną z zasad jest warstwowa architektura systemu. W interfejsach REST API wywołania i ich odpowiedzi przechodzą przez różne warstwy aplikacji. Z reguł projektowania wynika fakt, iż nie należy zakładać bezpośredniej komunikacji pomiędzy aplikacjami klienta i serwera. W celu komunikacji między aplikacjami używane są pętle komunikacyjne obejmujące pewną liczbę elementów bezpośrednich. W tak zaprojektowanej aplikacji, ani klient, ani serwer nie mają informacji o tym, czy komunikują się z aplikacją końcową czy elementem pośredniczącym.

Ostatnią opcjonalną zasadą jest zasada wysyłania kodu na żądanie. Oznacza to, że interfejsy REST API powinny wysyłać elementy statyczne, jednak w pewnych przypadkach dopuszcza się wysyłanie kodu wykonywalnego, jednak jego uruchomienie powinno być możliwe tylko w wypadku jawnego żądania.

2. SPOSÓB DZIAŁANIA INTERFEJSÓW REST API

Interfejsy REST API do komunikacji wykorzystują żądania HTTP. Dzięki temu wykonywane zostają standardowe funkcje bazy danych, takie jak tworzenie, odczytywanie, aktualizowanie i usuwanie rekordów w zasobach. Operacje te znane są jako CRUD. Operacje CRUD wykonujemy odpowiednio poprzez żądania - GET by pobrać rekord z bazy, POST w celu utworzenia nowego rekordu, PUT służący do aktualizacji rekordu oraz DELETE mające za zadanie usunięcie rekordu. W wywołaniach zapytań do API wykorzystane mogą zostać wszystkie metody HTTP. Idealnie zaprojektowany i stworzony interfejs REST API działa podobnie do serwisu WWW działającego w przeglądarce. Zasoby przesyłane wraz z zapytaniami lub w odpowiedzi na nie nazywamy reprezentacją zasobów. Informacje o zasobie dostarczone mogą być do klienta w wielu formatach, jednak najpopularniejszym jest format JSON. Swoją popularność zawdzięcza bardzo dobrej czytelności zarówno dla ludzi jak i maszyn, a ponadto jest niezależny od języka programowania. Parametry i nagłówki przekazywane w żądaniach są bardzo ważne w przypadku REST API, ponieważ zawierają one informacje identyfikujące, takie jak metadane, autoryzacje, identyfikatory, dane o buforowaniu czy też pliki cookie.

To co wyróżnia REST API to elastyczność tworzenia projektów, jednak w przypadku niedoświadczonych programistów lub złego projektu interfejsu, może się on okazać mało wydajny lub wcale nie działać. W celu zminimalizowania ryzyka wystąpienia takich problemów programiści stosują sprawdzone procedury w specyfikacjach takich interfejsów. Specyfikacja OAS (ang. OpenAPI Specification) ustanawia interfejs służący do opisywania interfejsu API w taki sposób, aby umożliwić programistom i aplikacjom poznanie i co za tym idzie zrozumienie jego parametrów i możliwości – w tym dostępnych punktów dostępowych, dozwolonych operacji na każdym punkcie końcowym, parametrów operacji, metod uwierzytelniania i innych. W najnowszej wersji specyfikacji znajdziemy narzędzia pozwalające na generowanie kodu pośredniczącego serwera i klienta w różnych językach programowania.

3. PODSTAWOWA AUTORYZACJA HTTP

Podstawowa autoryzacja HTTP opiera się na mechanizmie żądania i odpowiedzi. Serwer żąda od klienta informacji uwierzytelniających, na które składa się ID użytkownika oraz hasło. Dane uwierzytelniające przesyłane są do serwera

w nagłówku autoryzacyjnym w kodowaniu Base64. W przypadku wysłania przez klienta żądania nie zawierającego danych uwierzytelniających, a serwer spodziewa się ich dla tego konkretnego żądania, do użytkownika zostanie zwrócona odpowiedź o błędzie w uwierzytelnianiu i braku możliwości zwrócenia danych.

```
WWW-Authenticate: Basic realm="Our Site"
```

Listing 1. Format nagłówka dla podstawowego uwierzytelniania WWW-Authenticate

Przedstawiony w listingu 1 nagłówek uwierzytelniania WWW, zawiera atrybut realm, który identyfikuje zestaw zasobów, do których będzie miał zastosowanie przesyłany identyfikator i hasło. Przeglądarki mogą przechowywać dane autoryzacyjne dla każdego z zestawów zasobów, aby użytkownik nie musiał za każdym razem wpisywać danych na nowo.

```
Authorization: Basic userid:password
```

Listing .2 Format podstawowego nagłówka autoryzacyjnego

W momencie gdy podane zostaną wspomniane wcześniej dane uwierzytelniające, wysyłane jest żądanie z nagłówkiem autoryzacyjnym, którego struktura zaprezentowana została w listingu 2. Jak widać składa się on z słowa kluczowego „Basic” oraz identyfikatora i hasła użytkownika oddzielonych od siebie separatorem „:”. Ten sposób uwierzytelniania jest ustandaryzowany i łatwy do implementacji. Jednak problemem z nim związanym jest fakt, iż po uwierzytelnieniu użytkownika aplikacja musi pobrać hasło. Z perspektywy użytkownika nie jest możliwe określenie co aplikacja robi z hasłem. Oznacza to pełny dostęp aplikacji do konta bez możliwości cofnięcia tego dostępu innego niż zmiana hasła. W przypadku tego uwierzytelniania hasła to tokeny, które mogą zostać przechwycone przez atakującego, a prawdopodobnie nie zostanie to zauważone. Warto zaznaczyć, że nie jest w tym przypadku możliwe uwierzytelnianie wieloskładnikowe.

4. KLUCZE API

Uwierzytelnianie przy pomocy kluczy API to sposób uwierzytelniania, który nie odwołuje się do konkretnego użytkownika. Aplikacja klienta w każdym ze swoich żądań przesyła do API klucz uzyskany od aplikacji serwera. Dzięki temu klient zostaje zidentyfikowany, a po pomyślnej autoryzacji dane zostają mu zwrócone. Co ważne uwierzytelnianie poprzez klucz API nie jest ustandaryzowane, oznacza to, że każde API korzystające z tego sposobu ma swoją własną implementację. Ten sposób uwierzytelniania jest szczególnie popularny w przy-

padku aplikacji udostępniających dane i statystyki. Klucze API zostały stworzone w kilku celach. Pierwszym z nich jest zapobieganie anonimowych zapytań mających na celu zwiększyć niepotrzebny ruch na serwerze i obciążenie stron udostępniających dane. Kolejny cel to kontrola liczby wywołań API, w wielu darmowych API możemy się spotkać z dobowym lub miesięcznym limitem zapytań. Dodatkowo mogą być one wykorzystywane do identyfikacji pewnych wzorców użytkownika API, które mogą być kluczowe przy próbach rozwiązania problemów interfejsu API lub wykrywania złośliwej aktywności i zapobieganiu takowych w przyszłości.

```
https://adres_url_.domena/sciezka_do_danych?key=KLUCZ_API
```

Listing 3. Uwierzytelnianie za pomocą klucza API

Zgodnie z tym co zostało zaprezentowane w listingu 3, przy każdym wysyłanym żądaniu przez aplikację klienta wysyłany jako argument w adresie URL prowadzącym do potrzebnych danych, uzyskany wcześniej klucz dostępowy. Największą zaletą tego podejścia do uwierzytelniania dostępu do danych jest prostota i przejrzystość rozwiązania. Rozwiązanie to jest jednak niebezpieczne w przypadku większych aplikacji, szczególnie tych przesyłających dane wrażliwe między serwerami. Największym problemem używania kluczy API jest łatwość ich przechwycenia. W przypadku komunikacji serwer – serwer możliwe jest zabezpieczenie klucza przy pomocy TLS, czyli rozwinięcia protokołu SSL. Jednak to rozwiązanie może być zastosowane tylko w przypadku aplikacji działających po stronie serwera. W każdym innym scenariuszu użytkownika klucza API przez klienta, jest bardzo duże prawdopodobieństwo na jego przechwycenie. Każde żądanie URL zapisywane jest w logach przeglądarki. Aplikacje JavaScript często są bardziej otwarte niż powinny. Z tych powodów zalecane jest użytkowanie kluczy API tylko w przypadku aplikacji, które nie posiadają danych wrażliwych.

5. AUTORYZACJA ZA POMOCĄ TOKENU Z WYKORZYSTANIEM OAUTH 2.0

Autoryzacja oparta na tokenach oznacza, że wszystkie żądania kierowane do serwera muszą zawierać token autoryzacyjny jako potwierdzenie zezwolenia na dostęp do konkretnych danych. Obecnie najpopularniejszym protokołem używania tokenów jest OAuth 2.0.

Określa on mechanizmy, dzięki którym aplikacja żąda o dostęp do usług w imieniu użytkownika i otrzymać token jako dowód, że zgoda na dostęp została wyrażona. Najlepszym sposobem na wytłumaczenie działania tego sposobu, będzie demonstracja jednego z przypadków użycia. Załóżmy sytuację, w której użytkownik posiada konto w naszym serwisie służącym do zapisywania pomia-

rów uzyskanego prądu z kolektorów słonecznych. Użytkownik chce pozwolić zewnętrznej aplikacji na dostęp do odczytywania danych zapisywanych na naszym serwisie w celu wyrysowania grafów. Nasze API posiada możliwość wypisania danych gromadzonych przez użytkownika jednak w celu ich uzyskania, żądający musi w jakiś sposób uwierzytelnić swoje żądanie. W celu lepszego zaprezentowania działania podejścia OAuth założmy, że nasz serwis posiada specjalny serwer autoryzacyjny zwracający tokeny, które umożliwiając będą dostęp aplikacjom zewnętrznym na dostęp do danych w imieniu użytkowników. Użytkownik zostaje uwierzytelniony po podaniu odpowiednich danych dostępowych do konta. Po uwierzytelnieniu użytkownik uruchamia dostęp do swoich danych aplikacjom zewnętrznym.

```
GET /odczyt HTTP/1.1
Host api.odczyt.pl
Authorization: Bearer „token”
```

Listing 4. Użycie tokenu uwierzytelniającego przez aplikację zewnętrzną

W tym przypadku potrzebować będziemy dostępu do danych o odczytach z kolektorów, jest to ważne ponieważ system autoryzacyjny musi wiedzieć, o który z zasobów powinien zapytać. Po akceptacji dostępu do danych przez użytkownika, wystawiany jest token będący dowodem zezwolenia na pobieranie danych. Token przesyłany jest do aplikacji zewnętrznej. Od teraz aplikacja wewnętrzna może żądać od naszego interfejsu API przy użyciu tokena uwierzytelniającego danych znajdujących się na koncie użytkownika, który jest właścicielem konta do którego token jest przypisany.

Token zostaje przesłany wraz z żądaniem w nagłówku autoryzacyjnym poprzedzony słowem kluczowym „Bearer”. Po otrzymaniu żądania serwer sprawdza poprawność tokenu, a dane zostają zwrócone. Dzięki temu zewnętrzna aplikacja może uzyskać potrzebne jej dane przypisane do konkretnego użytkownika w momencie gdy ich potrzebuje.

Warto zwrócić w tym miejscu uwagę na to jak wygląda walidacja przesyłanego przez aplikację wewnętrzną tokenu. Odbywać się to może na dwa sposoby, jednym z nich jest zwrócenie referencji do danych tokenu lub zwrócenie danych bezpośrednio. W przypadku referencji token wysyłany jest do serwera autoryzacyjnego w celu walidacji i zwrócenia danych związanych z nim. Proces ten nazywamy introspekcją (ang. Introspection).

```
{
  "active": true,
  "usr": "Test",
  "client_id": "grafy",
  "scope": "odczytaj_dane_o_wytwarzanej_energii"
  ...
}
```

Listing 5. Przykład odpowiedzi zawierającej dane powiązane z przesłanym tokenem do serwera autoryzacyjnego

W podanej w listingu 5 przykładowej odpowiedzi obserwujemy, że użytkownik „Test” zezwolił aplikacji „grafy” na dostęp do jego konta, z zakresem (ang. scope) do odczytywania danych z wytwarzanej energii przez jego kolektory słoneczne. Bazując na tych informacjach, serwer może zdecydować o tym czy zezwolić czy odrzucić żądanie od zewnętrznej aplikacji. Oczywiście zwracane może być dużo więcej pól zawierających dużo bardziej szczegółowe dane na temat użytkownika w celu lepszej weryfikacji jego dostępu do konkretnych chronionych danych.

Drugim sposobem jest zwracanie danych bezpośrednio. Odbywa się to zwykle z użyciem formatu JSON Web Token (JWT). Nazywane są one tokenami strukturalnymi (ang. structured token) i przesyłane w nagłówkach autoryzacyjnych żądań zawierają w sobie wystarczająco danych, by na ich podstawie podjąć decyzję o zezwoleniu na dostęp do danych, aplikacji zewnętrznej.

6. OpenID CONNECT

Ten rodzaj protokołu uwierzytelniania również bazuje na OAuth 2.0. Używa on standardowego przesyłu danych dla REST API. Celem stworzenia OpenID było ułatwienie tworzenia prostych rzeczy oraz skompilowanych możliwymi do utworzenia. Protokół umożliwia programistom uwierzytelnianie użytkowników pomiędzy stronami internetowymi i aplikacjami bez potrzeby przechowywania i zarządzania plikami zawierającymi hasła. W przypadku tego protokołu używany przez niego OAuth 2.0 zostaje rozszerzony o dodatkowe uwierzytelnianie stowarzyszone (ang. federated authentication). Oznacza to możliwość logowania do jednego serwisu za pośrednictwem konta z innego. Przykładem może być tutaj logowanie do dowolnie wybranego serwisu poprzez konto portalu Facebook. Sam proces jest bardzo podobny do wcześniej opisanego w punkcie 4 przepływu danych z jednym wyjątkiem, którym jest ID Token. Dla porównania w standardzie OAuth 2.0 korzystamy z możliwości delegowanej autoryzacji (ang. Delegated Authorization). Oznacza to, że na podstawie danych z jednego konta chcemy uzyskać jakieś dane z drugiego powiązane z nim. Jako przykład posłużyć nam może chęć pobrania swojej listy znajomych z portalu Facebook do innej umożliwiającej nam to aplikacji.

```
{
  "jti": "00000-e420-1111-5555-ef31dgert",
  "exp": 24092022,
  "nbf": 0,
  "iat": 1583169992,
  "iss": "http://przykladowadomena/auth/",
  "aud": "uzytkownik",
  "sub": "c3c3755a-e499-4782-b119-a19bede0ace8",
  "typ": "Serialized-ID",
  "nonce": "02c0b033-bfac-4030-a317-c18aec3cb2db",
  "auth_time": 1583169991,
  "acr": "1",
  "session_state": "8b66127b-4474-41bd-8e36-5d18286df73f",
  "state_checker": "HZZmC4no-TEqiCf31Mk1MtONDqyxkk81ZXZCwANQb9Y",
  "name": "Jan Kowalski",
  "email": "jan.kowalski@email.com"
}
```

Listing 6. Przykładowa zawartość ID Token

Token ID przedstawiony w listingu 6 można nazwać dowodem osobistym w formacie JWT, który został podpisany przez dostawcę czyli OpenID.

ID Token zawiera w sobie potwierdzenie tożsamości użytkownika, określa organ wydający, jest generowany dla określonej grupy, może zawierać losowy ciąg służący do identyfikowania pochodzenia, może określać kiedy i jak użytkownik został uwierzytelniony. Dodatkowo posiada informację o czasie wydania i ważności, może zawierać imię, nazwisko oraz adres email posiadacza. Jak wcześniej zostało wspomniane jest również podpisany cyfrowo, dzięki czemu może zostać zweryfikowany. Ten typ uwierzytelniania użytkownika idealnie nadaje się do różnych mediów społecznościowych czy też aplikacji internetowych ponieważ dzięki niemu jednym kontem możemy logować się do wielu miejsc dzięki czemu użytkownik nie musi pamiętać różnych haseł oraz nazw kont. Wiąże się to oczywiście z niebezpieczeństwem związanym z atakami hakerskimi często wymierzonymi właśnie w konta na portalu Facebook.

Gdy konto zostanie przechwycone, atakujący może uzyskać dostęp do więcej niż jednej aplikacji internetowej.

7. PODSUMOWANIE

Niniejszy artykuł przybliży aspekty standardów uwierzytelniania i autoryzacji użytkowników w aplikacjach korzystających z REST API. Opisane zostały najpopularniejsze rozwiązania funkcjonujące w aplikacjach z których korzystamy na co dzień. Każde z podejść do autoryzacji i uwierzytelniania uwzględnione

w artykule ma swoje dobre strony i miejsca podatne na niebezpieczeństwo ataku. Istotnym jest, aby przy projektowaniu aplikacji wziąć pod uwagę, które z rozwiązań jest dla nas najlepsze. Metody uwierzytelniania i autoryzacji są stale udoskonalane, a najnowszym opisanym w artykule sposobem jest OpenID Connect.

LITERATURA

1. Reschke J., *The 'Basic' http Authentication Scheme* 2015
2. Richer J., Sanso A., *OAuth 2 in Action* 2017

ŹRÓDŁA INTERNETOWE

3. <https://www.ibm.com/docs/en/cics-ts/5.1?topic=concepts-http-basic-authentication> (dostęp: 25.04.2022).
4. <https://nordicapis.com/the-difference-between-http-auth-api-keys-and-oauth/> (dostęp 25.04.2022)
5. <https://openid.net/connect/faq/> (dostęp 25.04.2022)

Jakub PRZYSTASZ

dr. inż. Bartosz TRYBUS
opiekun naukowy

STEROWNIKI PARAWIRTUALIZOWANE W SYSTEMIE MicrOS

Artykuł opisuje implementację sterowników urządzeń parawirtualizowanych w systemie operacyjnym MicrOS, na przykładzie sterownika Virtio-Net. System operacyjny MicrOS jest projektem rozwijanym przez Studenckie Koło Naukowe Informatyków „KOD”, przeznaczonym na platformę x86. Niniejszy artykuł zawiera opis implementacji sterownika urządzenia parawirtualizowanego Virtio-Net oraz przedstawia zasady na jakich opiera się jego działanie. Artykuł zawiera fragmenty kodu źródłowego sterownika Virtio-Net dla systemu MicrOS.

Słowa kluczowe: system operacyjny, sterowniki parawirtualizowane, interfejs programistyczny, virtio.

WPROWADZENIE

Wirtualizacja jest powszechnie stosowanym narzędziem w informatyce, które pozwala efektywnie wdrażać rozwiązania do systemów informatycznych bez potrzeby instalacji nowego sprzętu. Cechą wirtualizacji jest izolacja uruchomionych aplikacji w ramach jednego systemu komputerowego. Izolacja wdrożonych rozwiązań pozwala zwiększyć bezpieczeństwo poszczególnych elementów systemu, ponadto dzięki zastosowaniu wirtualizacji możliwe staje się skalowanie oraz łatwa migracja już uruchomionych aplikacji na inną infrastrukturę sprzętową, co jest nie bez znaczenia w przypadku usług wysokiej dostępności (High Availability).

Wirtualizacja jest możliwa dzięki wprowadzeniu nowej warstwy abstrakcji pomiędzy sprzętem a środowiskiem uruchomieniowym (systemem operacyjnym, kontenerem), co jest okupione zwiększeniem ilości wykonywanych instrukcji przez CPU. Zwiększenie ilości wykonywanych operacji ma kluczowy wpływ na szybkość działania oraz responsywność aplikacji działających w środowisku emulowanym.

Rozwiązaniem problemu spadku wydajności aplikacji działających w środowisku wirtualnym są technologie takie jak Intel-VT oraz AMD-V. Procesory obsługujące powyższe technologie potrafią przełączać kontekst pomiędzy sys-

temem gospodarza a maszynami wirtualnymi i dla każdego z przypadków wykonywać kod z taką samą szybkością. Wykorzystanie Intel-VT lub odpowiednika AMD-V niesie jednak za sobą pewne ograniczenia. Aplikacje uruchamiane w maszynie wirtualnej muszą być skompilowane na ten sam zestaw instrukcji co zestaw instrukcji zaimplementowany w procesorze systemu gospodarza. W przypadku korzystania z aplikacji napisanych na inne platformy sprzętowe wymagana jest emulacja nie tylko peryferii, ale również procesora – co spowoduje spadek wydajności.

Emulacja peryferii systemu komputerowego (pamięć, dyski twarde, karty sieciowe, etc.), również może zostać rozwiązana na dwa sposoby: pełna wirtualizacja oraz parawirtualizacja. W przypadku pełnej wirtualizacji system gościa (rezydujący w maszynie wirtualnej) nie jest świadom tego, że jest uruchomiony na maszynie wirtualnej, pracuje w ten sam sposób jak gdyby był uruchomiony natywnie na sprzęcie. Pełna wirtualizacja wymaga programowej emulacji sprzętu, co w negatywny sposób wpływa na wydajność uruchamianych w środowisku wirtualnym aplikacji. Całkowicie odmiennym podejściem do komunikacji pomiędzy maszyną wirtualną a systemem nadzorcy jest parawirtualizacja, która polega na tym, że system gościa jest świadom tego, że jest uruchomiony w maszynie wirtualnej. Takie podejście zdejmuje z nadzorcy obowiązek programowej emulacji peryferii, zamiast tego informacje pomiędzy systemami wymieniane są za pośrednictwem interfejsu. VirtIO jest jednym z najpopularniejszych interfejsów do komunikacji pomiędzy maszyną wirtualną a systemem nadzorcy.

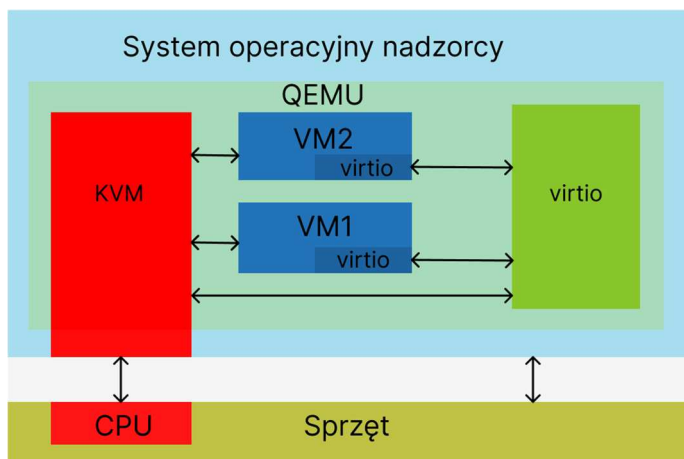
Zastosowanie parawirtualizacji możliwe jest tylko przy spełnieniu odpowiednich warunków:

- system nadzorcy udostępnia interfejs emulujący urządzenie
- system gościa posiada sterowniki obsługujące interfejs udostępniony przez nadzorcę.

Celem niniejszego artykułu jest przybliżenie sposobu przystosowania systemu MicrOS do pracy w środowisku parawirtualizowanym na przykładzie sterownika karty sieciowej Virtio-Net.

1. ŚRODOWISKO DO EMULACJI – QEMU

Środowisko Qemu pozwala na emulację kompletnego systemu, wraz z urządzeniami peryferyjnymi, obsługuje wiele architektur m.in. x86, x86_64, SPARC, AVR, MIPS, ARM. Ponadto umożliwia uruchomienie w trybie użytkownika aplikacji skompilowanych na inny zestaw instrukcji niż używany natywnie. Zaletą Qemu jest możliwość emulacji maszyny wirtualnej oraz wykonywanie jej instrukcji za pośrednictwem KVM lub Xen (z pominięciem programowej emulacji CPU korzystając z Intel-VT lub AMD-V), w takiej konfiguracji system gościa pracuje ze zbliżoną wydajnością do wydajności systemu nadzorcy. Dla peryferii emulator Qemu umożliwia parawirtualizację z wykorzystaniem interfejsu *virtio*, oraz emulację standardowych peryferii.



Rys. 1. Warstwy komunikacji pomiędzy maszyną wirtualną a sprzętem, opracowanie własne

Rysunek 1 przedstawia komunikację pomiędzy maszynami wirtualnymi Qemu a elementami systemu komputerowego, na którym pracuje emulator (nadzorca). Emulator udostępnia interfejs (urządzenie wirtualne *virtio*), z którego korzystają systemy operacyjne pracujące w maszynie wirtualnej, które to implementują sterowniki korzystające z udostępnionego im interfejsu. Zaletą zastosowania jednolitego standardu komunikacji jest to, że zmiana nadzorcy (np. z Qemu na lguest lub VirtualBox) nie spowoduje problemów w działaniu maszyn wirtualnych, natomiast wadą jest wymóg implementacji odpowiednich sterowników w systemie operacyjnym pracującym na maszynie wirtualnej. Na przykład system Windows domyślnie nie posiada zainstalowanych sterowników dla VirtIO, co wymaga ingerencji w maszynę wirtualną w celu osiągnięcia kompatybilności ze standardem *virtio*. W systemie operacyjnym MicrOS zaimplementowano sterownik do obsługi urządzenia sieciowego *virtio-net*.

2. WSPARCIE DLA PERYFERII

Specyfikacja *virtio* obecnie definiuje 10 rodzajów urządzeń peryferyjnych, tabela 1 zawiera listę urządzeń wraz z ich identyfikatorami zgodnie ze standardem *virtio*.

Tabela 1

Identyfikator	Nazwa
01	Karta sieciowa Ethernet (virio-net)
02	Urządzenie blokowe (dysk twardy) (virio-blk)
03	Konsola (virio-console)
04	Źródło entropii (virio-rng)
05	Memory Balloon (virio-balloon)
08	Kontroler SCSI (virio-scsi)
16	Adapter grafiki 2D/3D (virio-gpu)
18	Urządzenie wejścia (virio-input)
19	Gniazdo(socket) (virio-vsock)
20	Urządzenie kryptograficzne (virio-crypto)

Źródło: <https://docs.oasis-open.org/virtio/virtio/v1.1/cs01/virtio-v1.1-cs01.html>, dostęp 12.04.2022

Wartym uwagi urządzeniem jest „Memory Balloon”, które pozwala na wymianę informacji o zapotrzebowaniu na pamięć operacyjną przez maszynę wirtualną z nadzorcą (Qemu). Dzięki zaproponowanemu mechanizmowi nadzorca jest w stanie w dynamiczny sposób alokować pamięć dla maszyn wirtualnych w zależności od zapotrzebowania, co przekłada się na efektywniejsze wykorzystanie pamięci operacyjnej względem klasycznego podejścia, gdzie pamięć operacyjna jest stała dla każdej z maszyn wirtualnych. Przykładowo możliwe jest zadeklarowanie 16GB pamięci operacyjnej dla systemu gościa, po uruchomieniu systemu w maszynie wirtualnej zapotrzebowanie na pamięć może wynosić część z zadeklarowanej wcześniej pamięci np. 2GB. Za pośrednictwem sterownika memory balloon, system gościa jest w stanie powiadomić gospodarza o tym, że nie potrzebuje w danym momencie tyle pamięci, ile zostało mu przypisane. Niewykorzystana przez gościa pamięć trafia z powrotem do dyspozycji gospodarza, który może przydzielić ją np. innej maszynie wirtualnej. Jeżeli zapotrzebowanie na pamięć w systemie gościa wzrośnie, powiadomi on o potrzebie zwiększenia puli dostępnej pamięci za pośrednictwem omawianego mechanizmu, po czym zostanie mu przydzielona żądana ilość miejsca z puli dostępnej pamięci operacyjnej gospodarza. Niewykorzystana przez jedną maszynę wirtualną pamięć, może być jednocześnie przydzielona na potrzeby innej, co pozwala zredukować wymagania sprzętowe potrzebne do uruchamiania wielu wirtualnych środowisk jednocześnie.

Urządzenie virio-net emuluje działanie karty sieci Ethernet – pozwala na komunikację z maszyną wirtualną w standardzie IEEE 802. Standard virio-net opisuje jedynie sposób wymiany danych pomiędzy maszyną wirtualną a nadzor-

cą, to co stanie się z danymi gdy trafią do nadzorcy zależy od tego w jaki sposób został skonfigurowany. W przypadku QEMU istnieje możliwość połączenia maszyny wirtualnej z nadzorcą Point-To-Point (SLIRP) gdzie z perspektywy nadzorcy dostęp do maszyny wirtualnej jest możliwy poprzez jeden z portów TCP/UDP(komunikacja w warstwie transportowej). Możliwe jest również połączenie za pośrednictwem wirtualnego interfejsu sieciowego TUN(operuje na warstwie łącza danych ISO OSI RM), co daje możliwość dołączenia maszyny wirtualnej do fizycznej sieci IP.

3. KOMUNIKACJA Z PERYFERIAMI W ŚRODOWISKU WIRTUALNYM

Standard *virtio* określa trzy scenariusze komunikacji pomiędzy urządzeniem parawirtualizowanym a systemem operacyjnym gościa, są to:

- komunikacja z wykorzystaniem magistrali PCI
- komunikacja z wykorzystaniem mapowania rejestrów urządzenia do przestrzeni adresowej systemu operacyjnego (MMIO)
- komunikacja portami I/O dla maszyny S/390

Standardową metodą komunikacji z urządzeniami *virtio* jest komunikacja za pośrednictwem magistrali PCI, dalsza część artykułu skupiona będzie na tej metodzie wymiany danych. Każde urządzenie podłączone do magistrali PCI posiada nagłówek, w którym zawarte są podstawowe informacje na temat urządzenia, struktura takiego nagłówka została opisana w tabeli 2.

Tabela 2: Struktura zawierająca informacje o urządzeniu pracującym w magistrali PCI

Numer rejestru	Offset	Bity 31-24	Bity 23-16	Bity 15-8	Bity 7-0
00	00	Device ID		Vendor ID	
01	04	Status		Command	
02	08	Class code	Subclass	Prog IF	Revision ID
03	0C	BIST	Header type	Latency Timer	Cache Line Size
04	10	Base address #0 (BAR0)			
05	14	Base address #1 (BAR1)			
06	18	Base address #2 (BAR2)			
07	1C	Base address #3 (BAR3)			
08	20	Base address #4 (BAR4)			
09	24	Base address #5 (BAR5)			
0A	28	Cardbus CIS Pointer			
0B	2C	Subsystem ID		Subsystem Vendor ID	
0C	30	Expansion ROM base address			
0D	34	Reserved			Capabilities Pointer
0E	38	Reserved			
0F	3C	Max latency	Min Grant	Interrupt PIN	Interrupt Line

Źródło: <https://wiki.osdev.org/PCI> (dostęp 28.04.2022)

W implementacji sterowników kluczowe są pola Vendor ID, Device ID oraz Subsystem ID, których wartości zostały zdefiniowane w specyfikacji virtio, porównując wartości zapisane w tych polach możliwa jest identyfikacja urządzeń zgodnych ze standardem virtio. Standard urządzeń *virtio* określa następujące wartości dla pól w nagłówku PCI:

- Pole Vendor ID zawiera wartość równą 0x1AF4
- Pole Device ID wartości z przedziału 0x1000 do 0x107F
- Pole Subsystem ID określa rodzaj urządzenia zgodnie z identyfikatorem w tabeli 1

W systemie MicroS procedura rozpoznawania urządzeń połączonych magistralą PCI następuje w końcowej fazie sekwencji rozruchowej systemu. Kiedy system inicjuje sterowniki kart sieciowych odpytuje wszystkie urządzenia podłączone do magistrali PCI, w przypadku urządzenia *virtio-net* wyszukiwanie urządzenia w magistrali zostało pokazane na listingu 1:

```

bool virtio_nic_init(net_device_t *net_dev)
{
    // Prevent for re-init
    if (pci_virtio_nic_device.vendor_id != 0)
        return false;

    for (uint8_t i = 0; i < pci_get_number_of_devices(); i++)
    {
        pci_device *dev = pci_get_device(i);

        if (dev->vendor_id == VIRTIO_NET_DEVICE_VENDOR_ID &&
            dev->device_id >= VIRTIO_NET_DEVICE_ID_BEGIN &&
            dev->device_id <= VIRTIO_NET_DEVICE_ID_END &&
            dev->subsystem_id == VIRTIO_NET_DEVICE_SUBSYSTEM_ID)
        {
            pci_virtio_nic_device = *dev;
            break;
        }
    }

    // Device not present in PCI bus
    if (pci_virtio_nic_device.vendor_id == 0)
        return false;
    ...
}

```

Listing 1 Fragment funkcji sterownika virtio-net odpowiedzialny za wykrycie urządzenia w magistrali PCI

Jeżeli sterownik wykryje urządzenie pobiera zawartość pola *Base address 0*, odpowiednia interpretacja tego pola pozwala określić sposób komunikacji z urządzeniem. Tabela 3 oraz tabela 4 zawierają schematy w jakim należy interpretować zawartość pola *Base address 0*.

Tabela 3: BAR0 przy komunikacji za pomocą MMIO

Bity 31-4	Bit 3	Bity 2-1	Bit 0
Adres MMIO zapisany z wyrównaniem do 16 bajtów	Prefetchable	Typ	Zawsze 0

Źródło: <https://wiki.osdev.org/PCI>, dostęp 28.04.2022

Tabela 4: BAR0 przy komunikacji za pomocą portów IO

Bity 31-2	Bit 1	Bit 0
Adres portu I/O zapisany z wyrównaniem do 4 bajtów	Zarezerwowany	Zawsze 1

Źródło: <https://wiki.osdev.org/PCI>, dostęp 28.04.2022

W systemie MicrOS wspierana jest obecnie jedynie komunikacja za pośrednictwem portów wejścia/wyjścia, a pobranie adresu dla urządzenia *virtio-net* odbywa się w sposób zaprezentowany na listingu 2.

```
// Get device configuration address
virtio_nic.bar_type = pci_virtio_nic_device.base_addres_0 & (0x1);

if(!virtio_nic.bar_type)
// Driver can only configure device via IO ports.
    return false;
//Get IO address
virtio_nic.io_base = pci_virtio_nic_device.base_addres_0 & (~0x3);
```

Listing 2. Pobranie adresu portu IO w sterowniku virtio-net

4. KONFIGURACJA URZĄDZENIA

Pod pobranym adresem (*virtio_nic.io_base*) znajduje się struktura konfiguracyjna, której schemat jest wspólny dla każdego typu urządzenia zgodnego ze standardem virtio. Tabela 5 zawiera schemat takiej struktury.

Tabela 5: Struktura konfiguracyjna dla urządzeń virtio

Opis	Odczyt/Zapis (R/W)	Długość (bity)	Adres (względny)
Funkcje urządzenia	R	32	0x0
Funkcje sterownika	R+W	32	0x04
Adres kolejki	R+W	32	0x08
Rozmiar kolejki	R	16	0x0C
Wybór kolejki	R+W	16	0x0E
Numer następnego elementu w kolejce	R+W	16	0x10
Status urządzenia	R+W	8	0x12
Rejestr statusu przerw	R	8	0x13

Źródło: <http://docs.oasis-open.org/virtio/virtio/v1.1/virtio-v1.1.html> (dostęp: 10.04.2022)

Najważniejszym elementem powyższej struktury jest pole służące do kontroli stanu urządzenia, operacje zapisu/odczytu do tego pola pozwalają kontrolować stan urządzenia. Według specyfikacji virtio pole statusu może zawierać następujące wartości:

- ACKNOWLEDGE (1) – powiadomienie urządzenia, że sterownik poprawnie je wykrył
- DRIVER (2) – powiadomienie urządzenia, że sterownik jest w stanie z nim współpracować
- FAILED (128) – błąd wewnętrzny sterownika, dalsza komunikacja z urządzeniem niemożliwa
- FEATURES_OK(8) – powiadomienie urządzenia, że negocjacja wspieranych funkcji zakończona poprawnie
- DRIVER_OK(4) – sterownik gotowy do pracy z urządzeniem
- DEVICE_NEEDS_RESET(64) – błąd wewnętrzny urządzenia, wymagany restart

W systemie MicrOS sekwencja uruchamiająca urządzenie oraz negocjująca wymagane przez sterownik funkcje została zamieszczona w listingu 3.

Pola konfiguracyjne typowe dla konkretnego rodzaju urządzenia parawirtualizowanego zaczynają się od adresu 0x14 względem bazowego adresu urządzenia. Dla urządzeń *virtio-net* rejestry konfiguracyjne zostały opisane w tabeli 6.

```
// Reset the virtio-network device
virtio_nic_reg_write(REG_DEVICE_STATUS, STATUS_RESET_DEVICE);

// Set the acknowledge status bit
virtio_nic_reg_write(REG_DEVICE_STATUS, STATUS_DEVICE_ACKNOWLEDGE);

// Set the driver status bit
virtio_nic_reg_write(REG_DEVICE_STATUS, STATUS_DEVICE_ACKNOWLEDGE |
STATUS_DRIVER_LOADED);

// Read the feature bits
uint32_t features = virtio_nic_reg_read(REG_DEVICE_FEATURES);

// Make sure the features we need are supported
if ((features & REQUIRED_FEATURES) != REQUIRED_FEATURES)
{
    virtio_nic_reg_write(REG_DEVICE_STATUS, STATUS_DRIVER_FAILED);
    return;
}
// Tell the device what features we'll be using
virtio_nic_reg_write(REG_DRIVER_FEATURES, REQUIRED_FEATURES);
```

Listing 3. Sekwencja negocjująca funkcje sterownika virtio-net w systemie MicroOS

Tabela 6 Rejestry konfiguracyjne virtio-net

Adres	Pole
0x14	MAC Address 1
0x15	MAC Address 2
0x16	MAC Address 3
0x17	MAC Address 4
0x18	MAC Address 5
0x19	MAC Address 6
0x1A	Status
0x1B	Max Queue

Pola określające adres MAC dostępne są w każdym przypadku, natomiast pole statusu jest możliwe do odczytania przez sterownik jedynie wtedy gdy zostanie wynegocjowane włączenie funkcji *VIRTIO_NET_F_STATUS*, w tym polu dopuszczalne są dwie wartości:

- *VIRTIO_NET_S_LINK_UP* (0x1) – aktywne połączenie
- *VIRTIO_NET_S_ANNOUNCE* (0x2) – pojawienie się połączenia

Po tym gdy urządzenie *virtio-net* zostanie uruchomione należy przygotować pozostałe elementy systemu do współpracy z nowym urządzeniem są to:

- Magistrala PCI – włączenie opcji PCI Bust Mastering
- Sterownik przerw – włączenie wektora przerw

Realizacja powyższych czynności w sterowniku dla systemu MicrOS jest opisana w listingu 4.

```
// PCI bus mastering
uint32_t pci_bus_config = pci_io_in(&pci_virtio_nic_device, 1);
pci_bus_config |= 0x4;
pci_io_out(&pci_virtio_nic_device, 1, pci_bus_config);

// Enable IRQ
uint32_t irq_num = pci_virtio_nic_device.interrupt_line;
pic_enable_irq(irq_num);
idt_attach_interrupt_handler(irq_num, virtio_nic_irq_handler);
```

Listing 4. Konfiguracja peryferii w sterowniku virtio-net dla MicrOS

5. KOMUNIKACJA ZA POŚREDNICTWEM MECHANIZMU VIRTQUEUE

Standard Virtio wykorzystuje do komunikacji mechanizm kolejek tzw. Virtqueue, które są umieszczane w pamięci operacyjnej systemu gościa. W przypadku urządzenia virtio-net minimalna ilość skonfigurowanych kolejek to dwie: jedna do odbierania danych(RX), oraz jedna do wysyłania danych(TX) z urządzenia. Jeżeli sterownik wynegocjuje z urządzeniem funkcję *VIRTIO_NET_F_CTRL_VQ*, wymagane jest konfiguracja dodatkowej kolejki służącej do kontroli urządzenia. Ponadto możliwe jest wykorzystanie większej ilości par kolejek (RX,TX) jeżeli sterownik oraz urządzenie uzgodnią użycie funkcji *VIRTIO_NET_F_MQ*. Ilość par jest definiowana w rejestrze konfiguracyjnym urządzenia pod adresem 0x1B - Max queue[patrz tabela 6].

Każda z kolejek złożona jest z trzech obszarów:

- Tablica deskryptorów – wskaźniki na bufor y dla danych
- Driver area(avail) – przestrzeń dostępna dla sterownika
- Device area(used) – przestrzeń dostępna dla urządzenia

W systemie MicrOS element tablicy deskryptorów został zdefiniowany w następujący sposób:

```
typedef struct virtq_descriptor
{
    uint64_t address;
    uint32_t length;
    uint16_t flags;
    uint16_t next;
} virtq_descriptor;
```

Listing 5. Definicja deskryptora

Pole `address` w strukturze z listingu 5 zawiera wskaźnik na zaalokowany bufor, adres ten jest odczytywany przez urządzenie, z tego powodu zapisywany pod tym miejscem adres winien być adresem fizycznym.

W listingu 6 umieszczono fragment funkcji wypełniającej tablicę deskryptorów, zastosowanie makrodefinicji `DMA_ADDRESS_OFFSET` pozwala przetłumaczyć adres z przestrzeni wirtualnej jądra systemu na adres fizyczny. Przypisanie flagi `VIRTQ_DESC_F_DEVICE_WRITE_ONLY` mówi urządzeniu, że jest to deskryptor kolejki wejściowej (RX). Brak powyższej flagi określa deskryptor jako *tylko do odczytu* z perspektywy urządzenia, natomiast sterowniki widzi ten deskryptor jako dostępny do zapisu. Mechanizm flag dostępu pozwala uniknąć kolizji w wymianie informacji pomiędzy urządzeniem a sterownikiem.

```

for (uint16_t i = 0; i < buffers_count; ++i)
{
    uint32_t buffer = ((uint32_t)heap_kernel_alloc(VIRTIO_NET_MTU, 0) -
DMA_ADDRESS_OFFSET);

    virtqueue->descriptor_area[i].address = (uint64_t)buffer;

    virtqueue->descriptor_area[i].flags = VIRTQ_DESC_F_DEVICE_WRITE_ONLY;

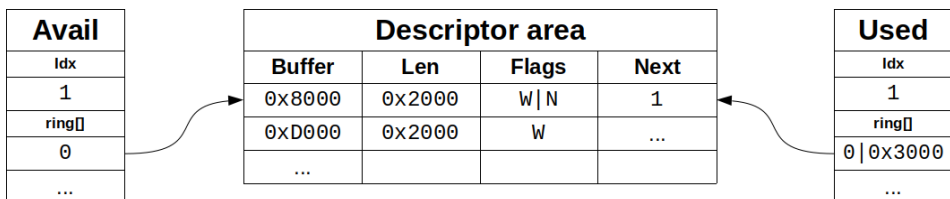
    virtqueue->descriptor_area[i].length = VIRTIO_NET_MTU;

    virtqueue->descriptor_area[i].next = 0;
}

```

Listing 6. Alokowanie buforów dla kolejki

W sytuacji gdy sterownik ma za zadanie wysłać dane do urządzenia, najpierw zapisuje bufory dostępne w tablicy deskryptorów, a następnie powiadamia urządzenie o nowych wpisach poprzez zmianę wartości index oraz ring w kolejce avail, po tym gdy urządzenie odczyta dane ze sterownika może powiadomić sterownik o poprawnym odczytaniu danych wpisując nowe wartości index oraz ring w kolejce used. Rysunek 2 zawiera diagram opisujący zasadę działania tablicy avail oraz used w kolejce TX urządzenia virtio-net. Jak pokazano na poniższym rysunku jeden z deskryptorów posiada flagę Next, która mówi urządzeniu, że przesyłane dane zostały rozłożone na dwa bufory, więc do odczytania pełnej informacji należy odebrać dane z następnego bufora. W tablicy used pod polem ring umieszczono strukturę wskazującą na pierwszy element z tablicy deskryptorów, oraz rozmiar odczytanych danych (0x3000).



Rysunek 2 Wymiana danych w kolejce wyjściowej (TX),

Źródło: <https://www.redhat.com/en/blog/virtqueues-and-virtio-ring-how-data-travels>
[dostęp 20.05.2022]

6. PODSUMOWANIE

Opisana w artykule implementacja sterownika dla virito-net realizuje podstawową funkcję wymiany ramek Ethernet z urządzeniem parawirtualizowanym z wykorzystaniem jedynie kilku funkcji oferowanych przez standard, rozbudowanie sterownika o obsługę kolejnych funkcji pozwoli zwiększyć efektywność wymiany informacji oraz niezawodność. Zrealizowanie funkcjonalności obsługujących komunikację za pośrednictwem virtqueue umożliwia realizację kolejnych sterowników dla pozostałych urządzeń parawirtualizowanych, ponieważ wszystkie zdefiniowane w standardzie virtio korzystają z tego samego mechanizmu.

LITERATURA

1. Russell, R.: Virtio: towards a de-facto standard for virtual I/O devices. SIGOPS Oper. Syst. Rev. 42 (2008) 95—103
2. Dandamudi S. P., Fundamentals of Computer Organization and Design, Springer, 2002.
3. Stallings W., Systemy operacyjne, architektura, funkcjonowanie i projektowanie, Helion, Gliwice 2018.

ŹRÓDŁA INTERNETOWE

4. <http://docs.oasis-open.org/virtio/virtio/v1.1/virtio-v1.1.html> (dostęp: 10.04.2022)
5. <https://developer.ibm.com/articles/l-virtio/> (dostęp: 10.04.2022)
6. <https://www.redhat.com/en/blog/virtqueues-and-virtio-ring-how-data-travels> (dostęp: 10.04.2022)

Oskar TYNIEC

dr inż. Bartosz TRYBUS
opiekun naukowy

ZASTOSOWANIE BIBLIOTEKI REACT NA PRZYKŁADZIE APLIKACJI DO ZARZĄDZANIA ZADANIAMI

Artykuł opisuje jedno z najbardziej nowoczesnych rozwiązań wykorzystywanych przy tworzeniu stron internetowych, jakim jest biblioteka React. Autor przedstawił w nim kluczowe elementy biblioteki takie jak komponenty, hooki czy rozszerzenie składni JavaScript – JSX. Dodatkowo wyjaśnione zostały formy połączenia z Rest API oraz koncept programistyczny znany szeroko jako Virtual DOM. W artykule zawarte zostało porównanie podobnych do tytułowego Reacta narzędzi programistycznych w postaci frameworków Vue.js oraz Angular. Ostatnim aspektem przedstawionym w artykule jest wykorzystanie biblioteki React do stworzenia aplikacji do zarządzania czasem, posiadającej system logowania z uwierzytelnianiem za pomocą tokena. Zwiercieniem tworzenia aplikacji jest stworzenie interfejsu użytkownika, który posłużyć może do prezentacji i obsługi przykładowych zadań w formie listy TODO.

Słowa kluczowe: react, frontend, biblioteka, framework, hook, komponent, virtual DOM.

WPROWADZENIE

Na świecie istnieje wiele rozwiązań programistycznych służących do tworzenia aplikacji internetowych. Dawniej głównymi narzędziami były czyste języki jak HTML, CSS, JavaScript oraz PHP. Współcześnie klasyczne interfejsy zostały zastąpione dynamicznymi stronami internetowymi zawierającymi wiele interaktywnych elementów oraz zależności. Zmiany te spowodowały, iż czyste języki programowania przestały być wystarczające w świecie nowoczesnych aplikacji. Z pomocą przyszły firmy takie jak Google, Facebook oraz niezależni programiści. Zauważywszy pojawiający się na rynku problem, stworzyli rozwiązania będące rozszerzeniem klasycznych języków programowania takie jak biblioteki i frameworki.

W artykule przedstawione zostaną trzy główne technologie używane we współczesnych aplikacjach. Są nimi:

- framework Vue.js,
- framework Angular,
- biblioteka React.

Rozwiązania te mimo wielu podobieństw, różnią się od siebie w wielu aspektach. Między innymi przeznaczeniem oraz poziomem wejścia w daną technologię. Porównanie wyżej wymienionych technologii zostanie opisane w punkcie „Porównanie narzędzi deweloperskich”. W poniższym artykule wykorzystane zostanie rozwiązanie stworzone przez firmę Facebook, jakim jest biblioteka React. Wybór danej technologii uzasadniony został w dalszej części artykułu. Narzędzie to zostanie wykorzystane w celu stworzenia prostej aplikacji do zarządzania zadaniami, co pozwoli na zaprezentowanie jego funkcjonalności oraz specyficznych aspektów.

1. PORÓWNANIE NARZĘDZI DEWELOPERSKICH

Każdy z deweloperów tworzących oprogramowanie staje przed niejednym wyborem. Pierwszym z nich jest wybór narzędzia, które posłuży do stworzenia aplikacji, spełniającej wymagania klienta. W tym celu warto zapoznać się z dostępnymi narzędziami, a następnie dobrać je pod tworzony produkt. W tabeli poniżej zaprezentowano podsumowanie dwóch frameworków: Vue.js, Angular oraz biblioteki React, które zostało rozwinięte w dalszej części artykułu.

	REACT	VUE	ANGULAR
Stworzony przez	Facebook	Evan You	Google
Popularność	Bardzo duża	Średnia	Średnia
Stabilność	Tak	Tak	Tak
Wsparcie wielkich firm / społeczności	Tak	Tak	Tak
Jakość dokumentacji	Dobra	Dobra	Średnia
Łatwość nauki	Średnia	Dobra	Średnia
Szybkość tworzenia kodu	Średnia	Dobra	Słaba

Rysunek 1. Tabela zawierająca porównanie frameworków
Źródło: opracowanie własne.

Pierwszym z omawianych narzędzi będzie Angular. Jest on najmłodszym z frameworków i najmniej popularnym z wymienionych technologii. Brak popularności zawdzięcza swojemu poprzednikowi AngularJS, który okazał się rozwiązaniem niespełniającym wymagań współczesnych programistów. Framework stworzony przez firmę Google w odróżnieniu od AngularJS wykorzystuje język TypeScript, co powoduje wiele trudności, między innymi potrzebę nauki nowego języka w celu skorzystania z rozwiązania. Kolejną wadą Angulara jest wiele niekompatybilnych ze sobą wersji. W połączeniu z ilością materiałów dostępnych w Internecie, w głównej mierze odnoszących się do AngularJS, powoduje trudności w nauce, jak również poszukiwaniu rozwiązania dla danego problemu. Technologia wykorzystywana przez firmy takie jak Udemy czy PayPal, świetnie sprawdza się w dużych aplikacjach, lecz jest zbyt wolna dla mniejszych serwisów.

Kolejnym frameworkiem, który został przedstawiony na rysunku 1, jest stworzony w 2014 roku, przez Evan'a You - Vue.js. W odróżnieniu od Angulara trzonem platformy jest język Java Script. Atutem Vue jest szybkość nauki, spowodowana dużą ilością materiałów internetowych oraz pozycji książkowych przedstawiających tę technologię. Vue jest najbardziej wydajne z porównywalnych platform, lecz rozmiar projektu jest największy. Powoduje to, iż prosta aplikacja napisana w Vue, jest zbyt obszerna jak na posiadane funkcjonalności. Framework jest idealnym rozwiązaniem do aplikacji średniej wielkości. Mimo prostoty oraz stabilności framework ten jest mniej popularny od biblioteki React.

Wyżej wspomniany React, w odróżnieniu od Vue oraz Angulara jest biblioteką. Została ona stworzona początkowo na potrzeby platformy społecznościowej przez firmę Facebook. Po pewnym czasie przekształcone w rozwiązanie otwarte dla społeczności React stał się najpopularniejszym z rozwiązań frontendowych. Wykorzystywany przez Netflix, Twittera czy Atlassian, jest idealny dla małych oraz średniej wielkości aplikacji. W kontekście nauki i wdrożenia programisty do biblioteki React pozwala na ułatwienie w postaci możliwości pisania kodu w Java Script lub TypeScript. Umożliwia to obniżenie progu wejścia dla początkujących programistów stron internetowych.

Mając na uwadze powyższą analizę dostępnych technologii frontendowych, należy przeanalizować wymagania produktu, który chcemy stworzyć. Aplikacja do zarządzania zadaniami jest małą aplikacją, zawierającą 3 podstawowe funkcjonalności. Są nimi: logowanie, wyświetlanie zadań oraz ich obsługa. Do tego wyzwania posłuży zatem biblioteka React, która jest najlepszym wyborem dla takiej aplikacji z racji swojej popularności i szybkości tworzenia kodu, jak również rozmiaru finalnego produktu.

2. TWORZENIE PROJEKTU

Wiele z nowoczesnych technologii programistycznych pozwala na stworzenie aplikacji przy pomocy kreatora. Tak jest i w przypadku biblioteki React, dla której to wystarczy użyć dedykowanego polecenia „create react app”. Polecenie to posiada jeden parametr, odpowiadający nazwie aplikacji, którą chcemy stworzyć. Zatem pierwszym krokiem do stworzenia aplikacji za pomocą zestawu narzędziowego Reacta jest wpisanie w wierszu poleceń frazy „npx create-react-app nazwa-aplikacji”. Proces tworzenia projektu za pomocą polecenia przedstawiony został na rysunku poniżej.

```
PS D:\> npx create-react-app todo-app

Creating a new React app in D:\todo-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

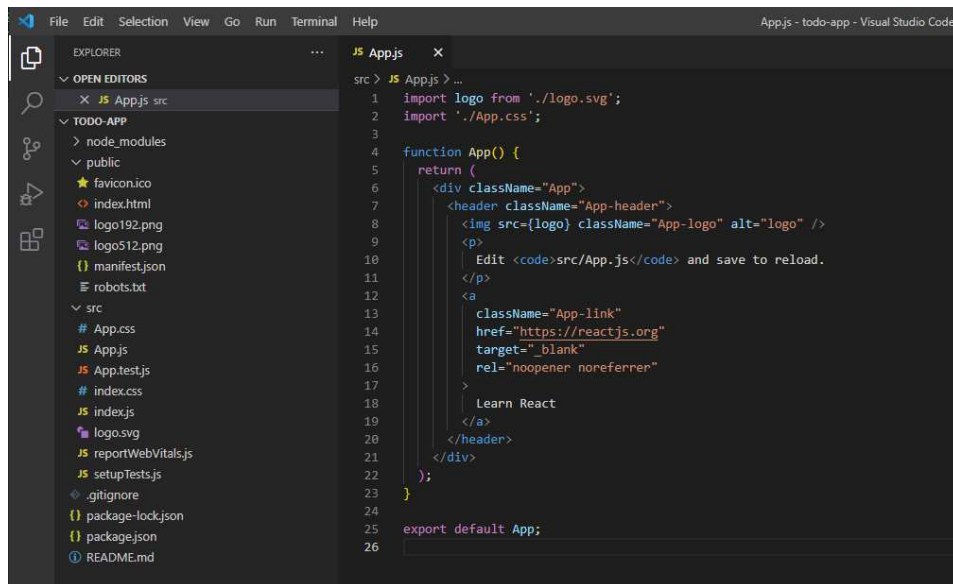
added 1352 packages in 1m

174 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.
```

Rysunek 2. Proces tworzenia projektu za pomocą zestawu narzędziowego
Źródło: opracowanie własne.

Po utworzeniu projektu możemy otworzyć go za pomocą środowiska programistycznego. Podczas pracy nad artykułem używane było środowisko Visual Studio Code. Jak można zauważyć na rysunku 3, utworzony projekt zawiera szereg gotowych elementów. Poza plikami z rozszerzeniem .js pojawiają się również gotowe arkusze stylów z domyślnymi własnościami dla utworzonych komponentów. Polecenie tworzenia projektu jest zatem dużym odciążeniem dla programisty, który bez pomocy zestawu narzędziowego, musiałby utworzyć całość projektu ręcznie.



Rysunek 3. Domyślne drzewo projektu stworzonego za pomocą polecenia create-react-app.

Źródło: opracowanie własne.

3. VIRTUAL DOM

DOM tłumaczony jako obiektowy model dokumentu, jest sposobem reprezentacji dokumentów HTML i XML w postaci obiektów znanych z wielu języków programowania. Obiekty te są budowane w pamięci przeglądarki podczas interpretowania kodu. DOM jest modelem programowania posiadającym wady, które w przypadku ciągle rozwijających się stron internetowych sprawiają pewne niedogodności. W dzisiejszych czasach, aplikacje i strony WWW są nastawione na dynamiczność oraz dużą ilość interaktywnych elementów wymagają częstych zmian. W modelu DOM wiąże się to z poważnym obciążeniem sprzętu, na którym renderowana jest strona. Naprzeciw temu problemowi wychodzi model Virtual DOM będący modyfikacją tradycyjnego modelu. Aplikacje wykorzystujące wirtualny model DOM wykonują renderowanie całej strony tylko podczas pierwszego uruchomienia strony. Każda zmiana w wyglądzie strony nie wymaga generowania strony od początku, a jedynie tworzy dany komponent strony i wgrywa go w już gotową stronę. Zaletą zastosowania Virtual DOM wykorzystywanego przez bibliotekę React jest zatem zwiększeniem wydajności stworzonej w tej technologii aplikacji.

4. JSX

Kolejnym z nieodłącznych elementów aplikacji tworzonych w bibliotece React jest zastosowanie JSX. Jest to rozszerzenie składni JavaScript pozwalająca na wstawienie znaczników HTML w kod Java Scriptu. W React nie ma sztucznego podziału logiki biznesowej od interfejsu użytkownika na osobne pliki. Powoduje to potrzebę połączenia wprost logiki i wyglądu w jednym pliku. Dlatego powstał JSX, pozwalający na płynne połączenie znaczników i mechanizmów stojących za działaniem strony internetowej. JSX jest co prawda opcjonalnym rozwiązaniem, lecz programiści korzystający z biblioteki React przyjęli je za pomocne rozwiązanie. Pomaga ono w prosty sposób wyświetlić interesujące nas dane oraz analizować błędy aplikacji w formie przejrzystego interfejsu graficznego. Dodatkowo JSX umożliwia pisanie kodu deklaratywnego, dużo prostszego w zrozumieniu niż kod imperatywny. Istotnym aspektem jest również wsparcie JSX dla nowoczesnych edytorów, jak i języka TypeScript. Jest to dużym ułatwieniem w porównaniu do pozostałych frameworków używających własnej składni szablonów.

```
src> JS TaskList.js >...
1  import { BiPencil, BiCheck } from 'react-icons/bi';
2
3  const TaskList = ({ tasks, onDelete, styles }) => {
4      return (
5          tasks.sort( (a,b) => {return b.priority - a.priority}),
6          tasks.map((task, index) => (
7
8              <div className={styles.task} key={index}>
9                  <div className={styles.editTask}>
10                     <div className={styles.editTaskIcon}>
11                         <BiPencil />
12                     </div>
13                 </div>
14                 <div className={styles.priority}>
15                     {task.priorityName}
16                 </div>
17
18                 <div className={styles.taskContent}> {task.content} </div>
19
20                 <button onClick={ () => onDelete({task}) } className={styles.acceptButton}><BiCheck /> Complete </button>
21             </div>
22         ))
23     );
24 }
25
26 export default TaskList;
```

Rysunek 4. Kod komponentu TaskList wykorzystującego rozszerzenie JSX
Źródło: opracowanie własne.

Wykorzystanie JSX zostało zobrazowane na rysunku 4. Dane takie jak priorytet zadania czy jego treść zostały dodane w kodzie Java Script za pomocą JSX.

5. KOMPONENTY

Kolejnym elementem, na który warto zwrócić uwagę podczas pisania aplikacji z użyciem frameworków frontendowych, są komponenty. React korzysta z nich w celu podziału interfejsu użytkownika na mniejsze części. Elementami tymi mogą być między innymi: przyciski, obrazy, formularze czy nawigacje. Komponenty są szczególnie przydatne w przypadku identycznych i powtarzających się w kilku miejscach elementów. Pozwalają one na zmniejszenie ilości pracy i napisanego kodu w trakcie tworzenia aplikacji poprzez wykorzystanie jednego komponentu oraz przekazaniu mu odpowiednich właściwości za pomocą parametrów. Tworząc komponenty należy zatem wziąć pod uwagę ich skalowalność i możliwość ponownego wykorzystania w innych miejscach aplikacji. Jest to ważne, aby komponenty miały swój sens, a nie zostały jedynie powielanymi elementami aplikacji pisanymi od nowa dla każdego możliwego zastosowania. Przykładem dobrze zaprojektowanego komponentu może być komponent przycisku. Przycisk może być wykorzystywany w wielu sytuacjach. Posługując się odpowiednio zaprojektowanymi danymi, przekazywanymi za pomocą właściwości (ang. Props) możemy wykorzystywać jeden komponent przycisku jako element odpowiedzialny za wysyłanie danych w formularzu, jak również ich usuwanie. Raz napisany komponent pozwala wtedy na wiele użyć w różnych zastosowaniach zamiast pisania kodu za każdym razem od nowa. Minusem wykorzystania komponentów jest brak z góry założonych wymogów i schematów dzielenia aplikacji na komponenty. Powoduje to niemałe problemy w przypadku pracy nad aplikacją w większych zespołach. W takich przypadkach należy założyć z góry podział elementów oraz ramy pracy z całym zespołem programistów.

6. HOOKI

Ciekawym elementem wyróżniającym Reacta spośród innych rozwiązań są Hooki. Hooki to funkcje pozwalające na obsługę stanu lokalnego komponentu (`useState`), obsługę cyklu życia komponentu (`useEffect`) oraz korzystanie z Context API poprzez hooka `useContext`. Hooki pozwalają więc na wykorzystywanie gotowych funkcji oraz pisanie własnych funkcji w formie hooków w celu współdzielenia zachowania między komponentami tworzonej aplikacji.

W aplikacji do zarządzania zadaniami użyte zostały dwa rodzaje hooków. Pierwszym z nich jest hook `useState`. Został on wykorzystany do obserwowania stanu danych umieszczanych przez użytkownika w polach formularza logowania. Inicjalizacja użytych w tym celu hooków została zaprezentowana na rys. 5.

W celu zachowania skalowalności, o której mowa była w punkcie zatytułowanym „Komponenty”, zostały wykorzystane uniwersalne nazwy zmiennych. Jak można zauważyć na rys. 6, komponenty zostały użyte w sposób pozwalający na wykorzystanie ich w innych miejscach. Komponenty przyjmują odpowiednie funkcje w formie pierwszej i drugiej wartości, a nie zaś zmiennej `login` i `hasła`.

```
const FormComponent = () => {  
  
  const [inputFirstValue, setFirstInputValue] = useState('');  
  const [inputSecondValue, setSecondInputValue] = useState('');  
  const [validateError, setValidateError] = useState(null);  
  
}
```

Rysunek 5. Kod komponentu FormComponent prezentujący inicjalizację hooków useState.
Źródło: opracowanie własne.

```
<p>  
  <span> Login </span>  
  <input onChange={handleChangeFirstInputValue} value={inputFirstValue}/>  
</p>  
<p>  
  <span> Password </span>  
  <input type="password" onChange={handleChangeSecondInputValue} value={inputSecondValue}/>  
</p>
```

Rysunek 6. Kod komponentu FormComponent prezentujący użycie komponentu
Źródło: opracowanie własne.

Do zmiany wartości stanów, wykorzystywanych później w logice odpowiadającej za logowanie użytkownika, posłużyły uchwyt zaprezentowane na rys. 7. Uchwyt te uzależnione od zmian treści formularza pozwalają na aktualizowanie zmiennych za pomocą hooków useState.

```
const handleChangeFirstInputValue = (event) => {  
  setFirstInputValue(event.target.value);  
}  
  
const handleChangeSecondInputValue = (event) => {  
  setSecondInputValue(event.target.value);  
}
```

Rysunek 7. Kod uchwytów odpowiadających za aktualizację zmiennych.
Źródło: opracowanie własne.

7. LOCAL STORAGE

Logowanie do aplikacji internetowych polega na przesłaniu danych logowania do serwera. Serwer po otrzymaniu danych powinien sprawdzić ich poprawność, a następnie odesłać aplikacji klucz dostępu do danych w formie access tokena. Za pomocą tego klucza aplikacja może następnie uzyskać dane z API, do których dostęp ma dany użytkownik oraz wykorzystywać je na własne potrzeby. W celu przechowania tokenu dla użytkownika naszej aplikacji użyty został magazyn danych w przeglądarce, nazywany Local Storage. Nie jest to co prawda najbezpieczniejsze rozwiązanie do przechowywania wrażliwych danych klienta, lecz na potrzeby prostej aplikacji testowej jest wystarczające. Dostęp do takiego magazynu jest możliwy za pomocą WEB Storage API naszej przeglądarki. Służą do tego metody:

- `setItem()`, za pomocą której możemy tworzyć oraz zmieniać wartość danych przechowywanych w formie klucz-wartość,
- `getItem()`, służąca do odczytywania wartości pozycji o podanym w parametrze kluczu,
- `removeItem()`, odpowiedzialna za usunięcie wskazanych danych.

Istnieje jeszcze kilka metod do obsługi Local Storage, lecz wyżej wymienione są najczęściej wykorzystywane w omawianym projekcie. W aplikacji do zarządzania zadaniami token odbierany po walidacji danych użytkownika jest zapisywany za pomocą `setItem()`. Prezentacja kodu odpowiadającego za obsługę zapisu tokena znajduje się na rys. 8.

8. POŁĄCZENIE Z REST API

Współczesne aplikacje internetowe korzystają z szerokiego zakresu narzędzi i systemów do efektywnego działania i przepływu danych. Jednym z nich jest API (ang. Application Programming Interface). Jest to kod zarządzający i obsługujący punkty dostępu do aplikacji nazywane endpointami. Strona logiki aplikacji programowanej na potrzeby artykułu, stworzona została za pomocą frameworka backendowego Laravel, wykorzystującego język PHP. Kod obsługujący zapytania został uruchomiony na darmowej platformie chmurowej Heroku. W celu integracji interfejsu aplikacji z serwerem należy utworzyć odpowiednie żądania HTTP. Pierwszym z żądań jest to odpowiadające za logowanie użytkownika. Strona serwera posiada dwie funkcjonalności, jedną z nich jest wspomniane logowanie, a drugą wyświetlanie oraz obsługa zadeklarowanych przez użytkownika zadań. Chcąc załogować się do aplikacji, należy wysłać specjalnie przygotowane dane na odpowiedni endpoint. W tym przypadku jest to endpoint `/api/token`. Pozwala on na sprawdzenie poprawności danych logowania użytkownika. Jeżeli dane zgadzają się ze stanem w bazie, następuje wysłanie tokena

dostępowego. W przypadku podania błędnych danych serwer odpowie konkretnym błędem, który w interfejsie użytkownika jest konwertowany na komunikat błędu. Przykładem takiego komunikatu jest „Incorrect data” wyświetlane podczas błędu walidacji formularza. Omawiany kod jest pokazany na rysunku 8.

```
async function handleSubmit(first, second){
  if(first !== '' && second !== ''){
    setValidateError('');

    const data = {
      email: first,
      password: second
    }

    const response = await fetch('https://enigmatic-atoll-16986.herokuapp.com/api/token/', {
      method: 'POST',
      headers: { "Content-Type": "application/json"},
      body: JSON.stringify(data)
    });

    if(response.ok) {
      const data = await response.json();

      if(response.status == 200){
        window.localStorage.setItem('access', data.access);
        window.localStorage.setItem('refresh', data.refresh);
      }
      else if(response.status == 401){
        setValidateError('Incorrect sign in data');
      }
      console.log(response.status);
    }
    else{
      setValidateError('Incorrect data');
    }
  }
  else{
    setValidateError('Incorrect sign in data');
  }

  getData();
}
```

Rysunek 8. Funkcja odpowiadająca za logikę logowania do aplikacji oraz zapis tokena
Źródło: opracowanie własne.

Po poprawnym zalogowaniu się użytkownika oraz otrzymaniu tokena dostępowego i zapisaniu go w Local Storage - następuje faza pobierania danych z serwera. W tym celu należy wysłać przygotowane wcześniej żądanie na endpoint /api/tasks. W przypadku zapytania dotyczącego pobrania danych korzysta się z metody GET. Dodatkowo należy pamiętać o umieszczeniu w nagłówku zapytania, tokena dostępowego otrzymanego w procesie logowania. Należy to zrobić tak jak na przykładzie zaprezentowanym na rys. 9.

```
async function getData(){
  console.log(window.localStorage.getItem('access'));

  const response = await fetch('https://enigmatic-atoll-16986.herokuapp.com/api/tasks', {
    method: 'GET',
    headers: { "Content-Type":"application/json", "Authorization": `Bearer ${window.localStorage.getItem('access')}` }
  });

  if(response.ok){
    console.log(response.json());
  }
}
```

Rysunek 9. Funkcja odpowiadająca za pobranie danych żądaniem GET
Źródło: opracowanie własne.

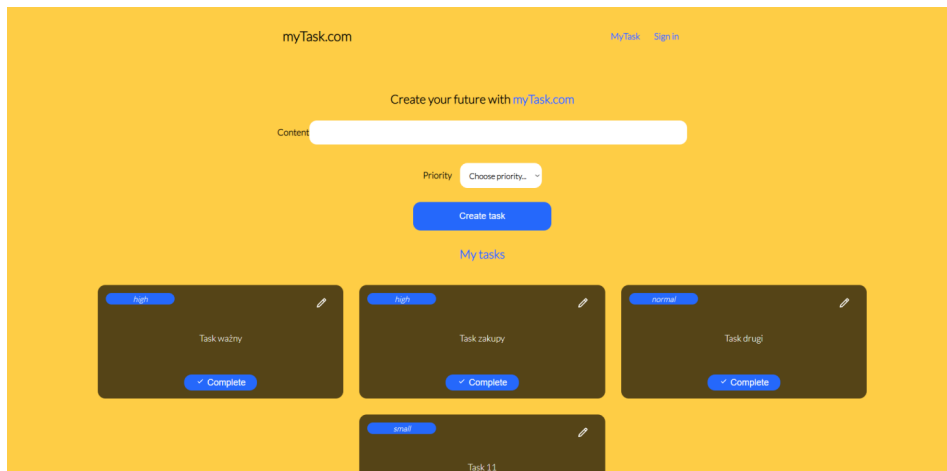
Dodając token do nagłówka żądania, warto pamiętać o określeniu jego typu. W tym przypadku jest to Bearer token, którego dołączono za pomocą metody `getItem()` odwołującej się do wartości wcześniej zapisanej w Local Storage. W bardziej rozbudowanych systemach używane są również tokeny zwane Refresh tokenem. Odpowiadają one za odnowienie tokenu dostępowego na chwilę przed jego wygaśnięciem. Token taki przydaje się w aplikacjach, w których użytkownik nie chce się uwierzytelniać przy każdym wejściu na stronę internetową.

Po przesłaniu żądania prezentowanego na rys. 9 należy odebrać odpowiedź serwera oraz zaprezentować pobrane dane użytkownikowi. Za prezentację otrzymanych danych odpowiada kod z rys. 10.

```
1 import { BiPencil, BiCheck } from 'react-icons/bi';
2
3 const TaskList = ({ tasks, onDelete, styles }) => {
4   return (
5     tasks.sort( (a,b) => {return b.priority - a.priority}),
6     tasks.map((task, index) => {
7
8       <div className={styles.task} key={index}>
9         <div className={styles.editTask}>
10          <div className={styles.editTaskIcon}>
11            <BiPencil />
12          </div>
13        </div>
14        <div className={styles.priority}>
15          {task.priorityName}
16        </div>
17
18        <div className={styles.taskContent}> {task.content} </div>
19
20        <button onClick={ () => onDelete({task}) } className={styles.acceptButton}><BiCheck /> Complete </button>
21      </div>
22    ))
23  );
24 }
25
26 export default TaskList;
```

Rysunek 10. Komponent odpowiadający za prezentację otrzymanych danych
Źródło: opracowanie własne.

Łącząc funkcjonalności wymienionych w artykule elementów tj. JSX, komponenty, hooki oraz języka programowania Java Script został stworzony powyższy kod. Pozwala on na posortowanie odebranych z serwera zadań w kolejności od największego do najmniejszego priorytetu. Dla zwiększenia przejrzystości danych zostały dodane również style CSS. Wygląd końcowej wersji aplikacji obrazuje rys. 11.



Rysunek 11. Wygląd aplikacji do zarządzania zadaniami po dodaniu logiki oraz stylów CSS
Źródło: opracowanie własne.

9. PODSUMOWANIE

Biblioteka React pozwala w prosty sposób stworzyć stronę internetową. Jest idealna dla małych oraz średnich produktów. W porównaniu do pozostałych rozwiązań frontedowych React nie wymaga od programisty pisania kodu w konkretnych ramach. Jest to z jednej strony uproszczeniem pracy, lecz podejście to ma swoje wady w przypadku pracy w zespołach. Jednak wciąż rosnąca popularność biblioteki omawianej w powyższym artykule skutkuje powstaniem wielu ogólnie dostępnych schematów pisania kodu, co niweluje ten problem.

LITERATURA

1. Kamiński P. , *React. Wstęp do programowania*. Wydawnictwo Helion, Listopar 2021,
2. Stefanov S. , *React w działaniu. Tworzenie aplikacji internetowych*, Wydawnictwo Helion, Kwiecień 2017.

ŹRÓDŁA INTERNETOWE

3. <https://github.com/facebook/react> (dostęp: 22.04.2022),
4. <https://github.com/skni-kod> (dostęp: 24.04.2022),
5. <https://pl.reactjs.org/docs/getting-started.html> (dostęp: 15.04.2022),
6. <https://reactjs.org/docs/getting-started.html> (dostęp: 13.04.2022).

