

	<p>Katedra Grafiki Wizji Komputerowej i Systemów Cyfrowych</p> <h2>Karta projektu SMiW</h2>		
Rok akademicki	Rodzaj studiów*: SSI Glw/SSI Ktw/NSI Glw	Dzień tygodnia i godzina zajęć:	Numer sekcji:
2024/2025	SSI Glw	Środa 8:00 – 9:30	12
Imię: Nazwisko:	Adrian Reszka	Prowadzący (inicjały): OA/JP/GB/KP	MP
<h3><i>Temat projektu</i></h3>			
<p>Inteligentny system monitorowania bezpieczeństwa domowego</p>			

1. Założenia projektu i opis funkcji urządzenia




Podstawowym zadaniem systemu będzie pomiar temperatury, wilgotności, a także wykrywanie dymu. Dodatkowo projekt będzie zawierał czujnik ruchu wykrywający intruza, aktywowany za pomocą osobnego przycisku. W sytuacji wystąpienia jakiegś nieprawidłowości – tzn. przekroczenie zakresów bezpiecznych dla pomiarów lub wykrycia ruchu, zostanie uruchomiony alarm i zaświeci się dioda LED, a użytkownik dostanie powiadomienie na podany przez siebie numer telefonu. Pomiarzy zebrane przez czujniki będą pokazywane na wyświetlaczu tak aby użytkownik miał do nich stały dostęp. W czasie tworzenia projektu zdecydowano się na dodanie dwóch dodatkowych przycisków. Jeden z nich służy do wyłączenia alarmu, natomiast drugi do zresetowania obecnych ustawień służących do łączenia się z siecią i wysyłania wiadomości SMS.

2. Elementy potrzebne do realizacji projektu

- mikrokontroler
- czujnik temperatury i wilgotności
- czujnik dymu
- czujnik ruchu
- dioda LED
- przyciski do aktywacji czujników
- ekran do wyświetlania pomiarów

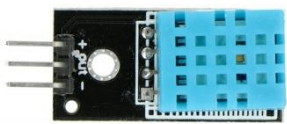

3. Przegląd dostępnych na rynku elementów wraz z ich uzasadnieniem wyboru

3.1 Wybór mikrokontrolera

Nazwa	ESP32 WiFi	ESP32 – C3	Arduino Nano
Zdjęcie			
Cena	49,90 zł	29,90 zł	116,00 zł
Napięcie zasilania	5V	5V	7V – 12 V
Porty I/O	34	15 pinów I/O	22 piny I/O
Prąd na I/O	Do 80mA	Od 50 do 100mA	Do 40mA

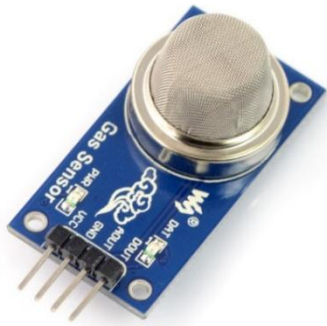
Wybrałem moduł ESP32 WiFi ze względu na bardzo dużą ilość pinów, co będzie kluczowe przy realizacji zadania. Moduł ESP32 – C3 właśnie przez liczbę złącz nie będzie odpowiednim komponentem do oparcia na nim projektu. Natomiast Arduino Nano posiada bardzo zbliżone możliwości do ESP32 WROOM jednakże jego cena jest wręcz zaborcza, a poza tym posiada mniejszą pamięć i nie posiada modułu WiFi.

3.2 Wybór czujnika temperatury i wilgoci

Nazwa	DHT11	AM2320 - I2C
Zdjęcie		
Cena	13,90 zł	19,90 zł
Zakres pomiarowy	Temperatura 0-50 °C ±1°C Wilgotność 20-90 %RH ±4 RH	Temperatura od - 40 °C do 80 °C ±0,5°C Wilgotność od 0 % do 99,9 % RH ±3 RH
Napięcie/Prąd	3V – 5,5V / 0,2mA	3,3V – 5,5V / 0,35mA
Ilość pinów	3	4


Wybrałem moduł czujnika DHT11, korzystający z interfejsu cyfrowego, ze względu na jego wystarczający dla pomieszczeń domowych zakres pomiaru temperatury i wilgotności. Nie ma sensu dopłacać do większej dokładności jeżeli jej nie potrzebujemy.

3.3 Czujnik dymu

Nazwa	MQ – 135
Zdjęcia	
Cena	23,90 zł
Napięcie / Prąd	2,5V – 5V / 135mA
Ilość pinów	4


Przy wyborze czujnika dymu nie było za wiele możliwości. Ich stanowcza większość pracuje na stałym napięciu zasilania 5V. Wybór ESP32 jako mikrokontrolera w pewien sposób narzuca wybór czujników zasilanych napięciem 3,3V, a wybrany przeze mnie czujnik pracuje w zakresie od 2,5V do 5V. MQ – 135 może korzystać z interfejsu analogowego jak i cyfrowego.

3.4 Czujnik ruchu

Nazwa	Cyfrowy czujnik ruchu PIR
Zdjęcie	
Cena	20,90 zł
Napięcie / Prąd	3V – 5V / 50uA
Ilość pinów	3

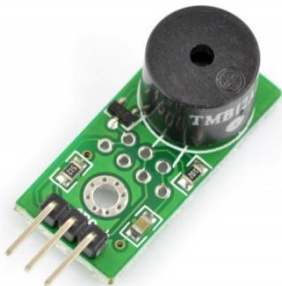
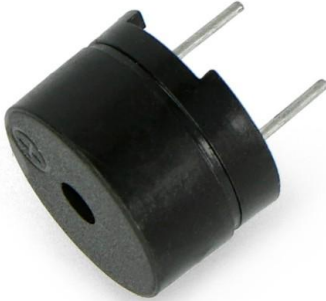
Tutaj również nie było dużego wyboru. Czujniki ruchu PIR są przystosowane do pracy na napięciu zasilania oscylującego w okolicach 5V, a co za tym idzie nie będą pasowały. Udało mi się znaleźć jeden moduł, który pracuje na napięciu 3,3V i dlatego też go wybrałem. Będzie on pracował na interfejsie cyfrowym.

3.5 Diody LED

Nazwa	Diody LED
Zdjęcie	
Cena	2,90 zł
Napięcie / Prąd	2V – 2,3V / 20mA
Ilość pinów	2

Wybór diod LED był dość prosty – wybrałem najtańszą z dostępnych opcji. Zależy mi jedynie na tym aby świeciła ona zauważalnym zarówno w dzień jak i w nocy światłem.

3.6 Buzzer

Nazwa	Moduł Buzzera z generatorem	Buzzer bez generatora
Zdjęcie		
Cena	5,50 zł	3,40zł
Napięcie / Prąd	3V – 5V / 20 mA	3V – 16V / 7 mA
Ilość pinów	3	2


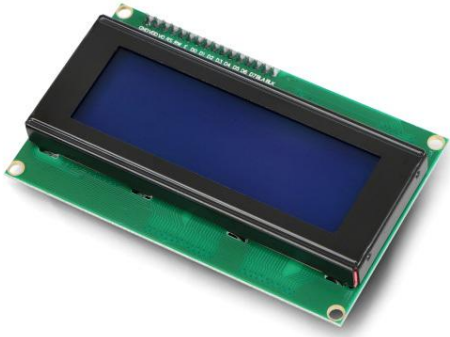
Mój wybór padł na moduł Buzzera z generatorem, ponieważ zależy mi na jednej, stałej częstotliwości. Dzięki temu układ będzie powiadamiał użytkownika o niebezpieczeństwie. Układ z Buzzerem aktywnym nie wymaga napisania kodu, gdyż generator w nim zawarty generuje dźwięk autonomicznie. Moduł korzysta z interfejsu cyfrowego.

3.7 Przyciski

Nazwa	Tact Switch 5szt. SWT-11138
Zdjęcie	
Cena	2,90 zł
Ilość pinów	2


Przy wyborze przycisków było wiele możliwości jednakże większość z nich różni się jedynie wyglądem i ceną. Wybrałem takie, które prezentują się dość oryginalnie oraz są przystępne cenowo.

3.8 Wyświetlacz

Nazwa	OLED 0,96" 128x64px I2C (sterownik zgodny z SSD1306)	LCD 4x20 + konwerter I2C (sterownik zgodny z HD44780)
Zdjęcie		
Cena	14,90 zł	34,90 zł
Napięcie / Prąd	3,3V lub 5V / 30 mA	5 V / 50 mA
Ilość pinów	4	4

Wybrałem wyświetlacz OLED, pracujący na interfejsie I2C ze względu na lepszą jakość obrazu i sam fakt iż nigdy nie pracowałem z takim ekranem, a chciałbym spróbować czegoś nowego. W przypadku wyświetlacza LCD musielibyśmy dopłacić do większego rozmiaru ze względu na ilość pomiarów do wyświetlenia, a także konwerter I2C, który znacznie zmniejszy ilość potrzebnych do zamontowania go przewodów.

3.9 Zasilanie

Nazwa	Powerbank CELLULARLINE Essence 20000 mAh 12W
Zdjęcie	
Cena	109,99 zł
Prąd	2,4 A

Postanowiłem, że układ będzie zasilany za pomocą Powerbanka. Wytwarzają one stałe napięcie 5V na wyjściu co będzie idealne do wybranego przeze mnie modułu zawierającego ESP32. Podałem tutaj przykładowy, jednakże posiadam kilka w domu i to na nich będę testował moduł. Nie mają one dużych pojemności (do 10000mAh), ale wraz ze wzrostem budżetu i upływu czasu zaopatrzę się w coś większego. Na samym początku brałem również pod uwagę zasilanie bateryjne jednakże jest to rozwiązanie które nie da takiej wydajności i wymaga systematycznej wymiany w przeciwieństwie do Powerbanka.

3.10 Wysyłanie sygnału użytkownikowi

Ostatecznie zdecydowałem się na wysyłanie powiadomienia o zagrożeniu przy pomocy aplikacji WhatsApp oraz dostępnego za darmo bota CallMeBot API. Zostanie użyty do tego moduł WiFi zamontowany w wybranym przeze mnie mikrokontrolerze. Użytkownik po połączeniu się z Access Pointem wygenerowanym przez mikrokontroler wejdzie na stronę z gotowym formularzem HTML aby podać potrzebne dane. Układ zresetuje się, połączy się z siecią i będzie gotowy do wysyłania powiadomień.

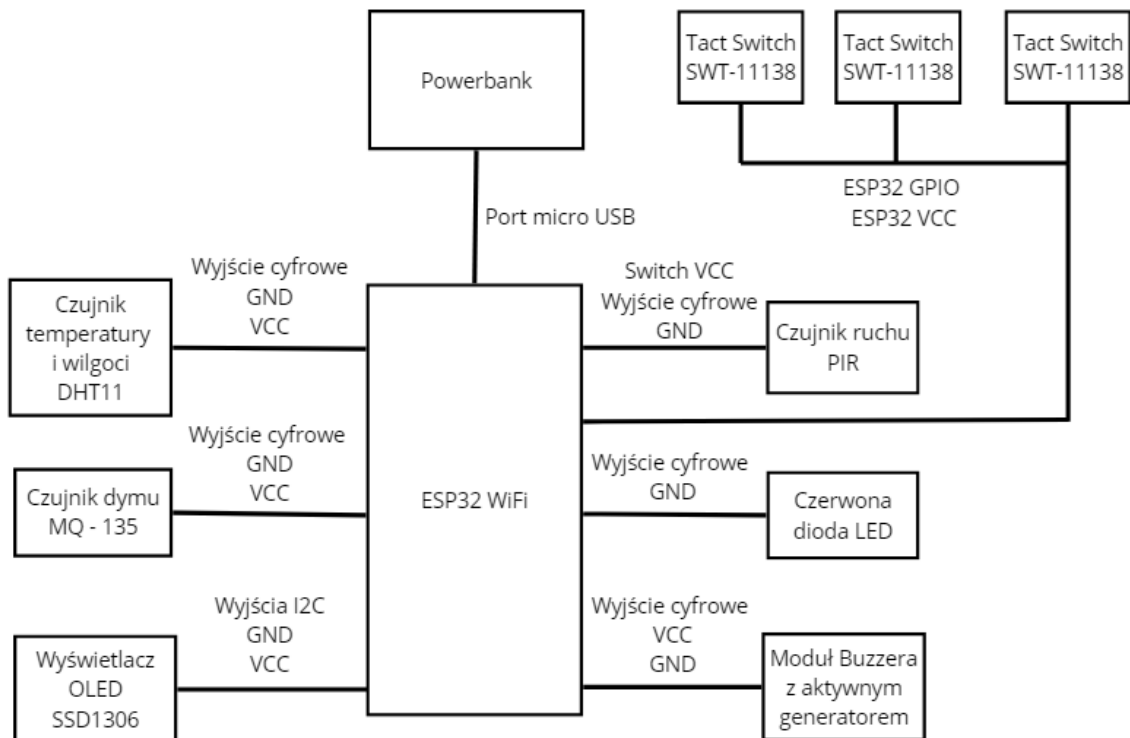
3.11 Kosztorys

Element	Ilość	Cena
Mikrokontroler ESP32 WiFi	1	49,90
Czujnik temperatury i wilgoci DHT11	1	13,90
Czujnik tlenku węgla i łatwopalnych gazów MQ - 135	1	23,90
Czujnik ruchu PIR	1	20,90
Diody LED	16	2,90
Moduł Buzzera z generatorem	1	5,50
Przyciski	5	2,90
Wyświetlacz I2C SSD1306	1	14,90
Zasilanie - Powerbank (opcjonalne)	1	109,99
Suma	28 (27)	244,79 (134,80)

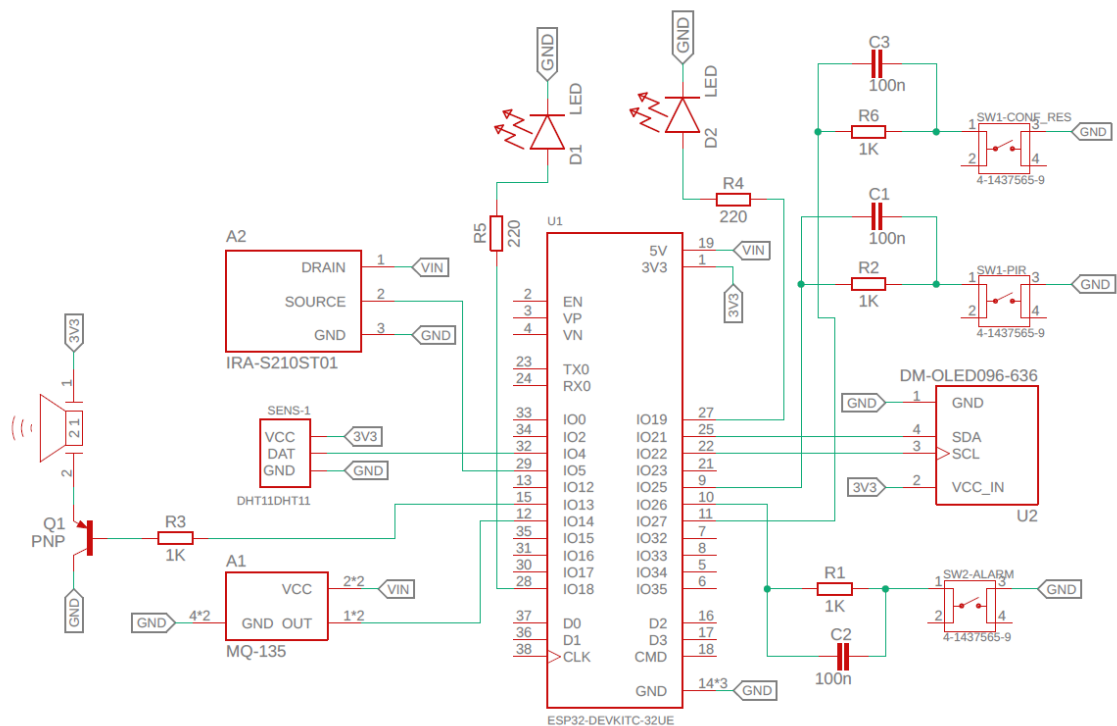
Trzeba zaznaczyć iż cena w nawiasie nie uwzględnia zakupu Powerbanka.

4. Specyfikacja wewnętrzna urządzenia

4.1 Schemat blokowy



4.2 Schemat ideowy



4.3 Opis elementów układu

4.3.1 Moduł ESP32 Wi-Fi

ESP32 to wszechstronny mikrokontroler wyposażony w łączność Wi-Fi oraz Bluetooth. Dzięki temu umożliwia zdalne monitorowanie i sterowanie systemem. Posiada dwa rdzenie oraz częstotliwość taktowania do 240 MHz, co zapewnia dużą wydajność. Wyposażony jest w liczne porty cyfrowe i analogowe, interfejsy takie jak UART, I2C czy SPI, co pozwala na podłączenie wielu urządzeń peryferyjnych.

4.3.2 Powerbank

Powerbank pełni rolę źródła zasilania układu, dostarczając napięcie 5 V poprzez port USB. Zapewnia mobilność systemu i niezależność od stałego zasilania sieciowego, co sprawia, że układ może działać w dowolnym miejscu.

4.3.3 Moduł czujnika temperatury i wilgotności DHT11

Czujnik DHT11 mierzy temperaturę w zakresie od 0 °C do +50 °C z dokładnością ± 1 °C oraz wilgotność w zakresie 20–90% RH z dokładnością $\pm 4\%$ RH. Posiada własny wbudowany protokół jednokierunkowy. Do komunikacji z mikrokontrolerem wykorzystuje pojedynczy pin cyfrowy. Jego prostota i niski koszt sprawiają, że jest idealnym rozwiązaniem do podstawowych aplikacji monitorowania środowiska.

4.3.4 Czujnik dymu MQ-135

Czujnik MQ-135 wykrywa obecność gazów takich jak dym, amoniak, benzen, dwutlenek węgla i inne szkodliwe związki w zakresie od 10 ppm do 1000 ppm. Do komunikacji z mikrokontrolerem wykorzystuje pin analogowy, co pozwala na pomiar proporcjonalnego napięcia do stężenia gazu. Działa na napięciu od 2,5 V do 5 V i jest kluczowym elementem układu alarmującego o obecności szkodliwych substancji w otoczeniu.

4.3.5 Wyświetlacz OLED SSD1306

Wyświetlacz OLED z wbudowanym sterownikiem SSD1306 umożliwia wyświetlanie tekstu i grafiki na niewielkim ekranie. Pracuje z napięciem 5 V i komunikuje się z mikrokontrolerem poprzez interfejs I2C, co upraszcza okablowanie i pozwala na podłączenie innych urządzeń korzystających z tego samego interfejsu. Jego wysoka kontrastowość i niski pobór mocy są idealne do zastosowań przenośnych.

4.3.6 Czujnik ruchu PIR

Czujnik ruchu PIR wykrywa ruch w otoczeniu, mierząc zmiany promieniowania podczerwonego. Działa na napięciu od 3 V do 5 V i generuje sygnał cyfrowy, który informuje o obecności obiektu w polu widzenia czujnika. Jest często stosowany w systemach alarmowych i automatyki domowej.

4.3.7 Moduł buzzera z aktywnym generatorem

Moduł buzzera generuje dźwięk alarmowy o stałej częstotliwości, informując użytkownika o zdarzeniach, takich jak wykrycie dymu czy ruchu. Pracuje na napięciu od 3 V do 5 V, a jego kompaktowy rozmiar umożliwia łatwą integrację z systemem.

4.3.8 Przyciski

W układzie zastosowano trzy przyciski o różnych funkcjach:

- Włączanie/Wyłączanie czujnika PIR – umożliwia aktywację lub dezaktywację detekcji ruchu
- Wyłączanie alarmu – pozwala na manualne wyłączenie sygnału dźwiękowego
- Resetowanie ustawień Access Pointa ESP – przywraca domyślne ustawienia sieciowe, ułatwiając ponowną konfigurację

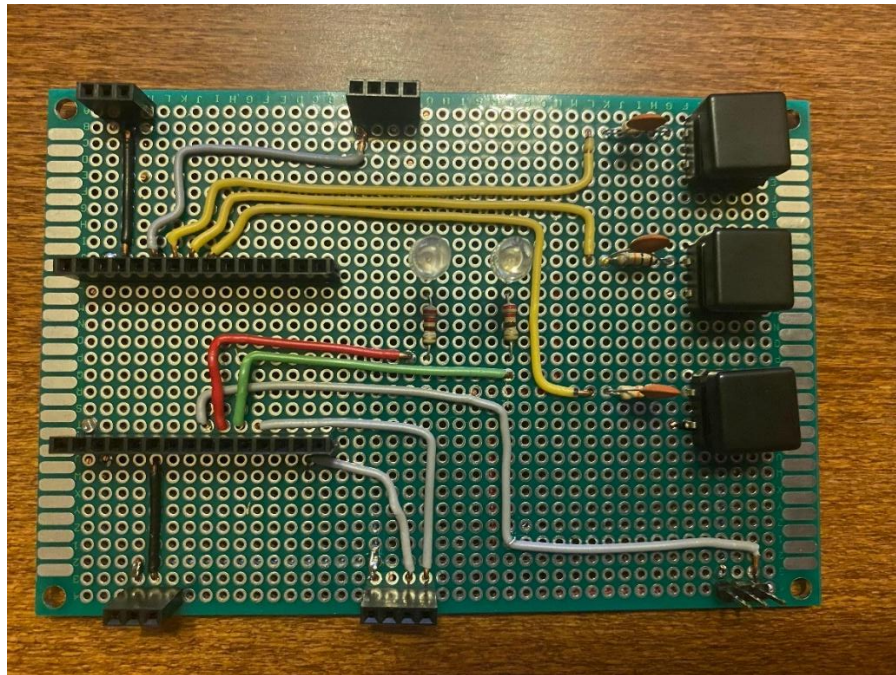
4.3.9 Elementy dodatkowe

W projekcie wykorzystano również typowe komponenty elektroniczne, takie jak:

- Rezystory – do ograniczania prądu
- Kondensatory – do tłumienia zakłóceń w obwodach
- Diody LED – sygnalizujące pracę układu

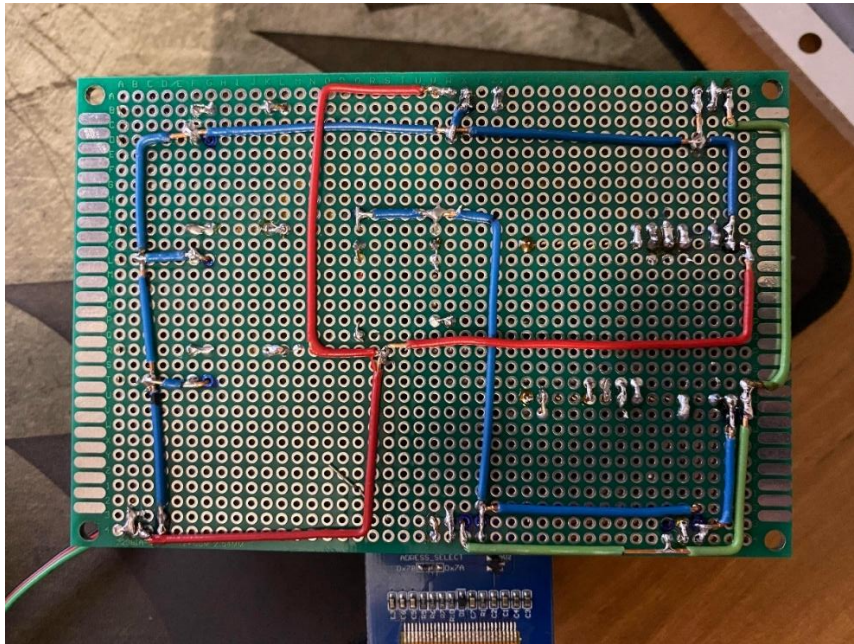
4.4 Schemat montażowy projektu

4.4.1 Widok na górną część uniwersalnej płytki PCB



Wlutowane gniazda i piny służą odpowiednio do zamontowania następujących elementów układu – licząc od lewego górnego rogu i przechodząc na dół – moduł Buzzera, czujnika MQ – 135, modułu czujnika DHT11, wyświetlacza OLED oraz czujnika ruchu PIR. Gniazda znajdujące się na środku lewej krawędzi płytki są przeznaczone do zamontowania w nich mikrokontrolera ESP32.

4.4.2 Widok na dolną część uniwersalnej płytki PCB



Przewody niebieskie zostały poprowadzone do masy, natomiast czerwone i zielone to zasilanie.

4.5 Algorytm programowania urządzenia

Cały kod został napisany w środowisku Arduino IDE. Komunikacja z mikrokontrolerem odbywała się poprzez kabel USB podłączony do komputera. Program został napisany w języku C++, który jest przystosowany do projektowania takich urządzeń. Zostały zastosowane zewnętrzne biblioteki, które pomogły w programowaniu takich elementów jak na przykład wyświetlacz czy czujnik DHT11.

4.5.1 Kod programu

Całość kodu została umieszczona w poniższym repozytorium na platformie GitHub:

- <https://github.com/AdrianReszka/SBD-SMIW-Project>

4.5.2 Opis zmiennych globalnych

display – Obiekt klasy Adafruit_SH1106G reprezentujący wyświetlacz OLED, używany do wyświetlania danych i komunikatów

dht – Obiekt klasy DHT do obsługi czujnika temperatury i wilgotności DHT11

sensorEnabled – Zmienna logiczna, informująca, czy czujnik PIR jest aktywny

alarmPIREnabled – Zmienna logiczna, sygnalizująca, czy alarm wywołany przez czujnik PIR jest włączony

alarmMQEnabled – Zmienna logiczna, sygnalizująca, czy alarm wywołany przez czujnik MQ-135 jest włączony

sensorStateMQ – Zmienna przechowująca aktualny stan cyfrowego wejścia dla czujnika MQ-135

activationStartTime – Zmienna unsigned long przechowująca czas aktywacji czujnika PIR

pirReady – Zmienna logiczna, wskazująca, czy czujnik PIR jest gotowy do pracy po upływie czasu aktywacji

pirMessageSent – Zmienna logiczna, wskazująca, czy wysłano wiadomość o ruchu wykrytym przez PIR

gasMessageSent – Zmienna logiczna, wskazująca, czy wysłano wiadomość o wycieku gazu

tempHumMessageSent – Zmienna logiczna, wskazująca, czy wysłano wiadomość o wysokiej temperaturze lub wilgotności

preferences – Obiekt klasy Preferences, używany do przechowywania konfiguracji w pamięci nieulotnej ESP32

ssid – String, przechowuje SSID sieci Wi-Fi

password – String, przechowuje hasło do sieci Wi-Fi

phoneNumber – String, przechowuje numer telefonu do wysyłania powiadomień

apiKey – String, przechowuje klucz API dla usługi powiadomień

server – Obiekt klasy WebServer, obsługujący serwer HTTP na ESP32

dnsServer – Obiekt klasy DNSServer, obsługujący serwer DNS dla trybu Access Point

5.4.3 Opis działania wszystkich procedur

setup() – konfiguruje wszystkie komponenty, takie jak wyświetlacz, czujniki, przyciski oraz sieć Wi-Fi. Ustawia piny jako wejścia.

loop() – główna pętla programu. Obsługuje logikę działania układu, przetwarza dane z czujników, steruje alarmami i wyświetlaczem oraz obsługuje żądania serwera HTTP.

displayData() – wyświetla aktualne dane dotyczące temperatury i wilgotności na wyświetlaczu OLED

displayMessage(const char *message) – wyświetla na ekranie OLED pojedynczy komunikat tekstowy. Jako argument przyjmuje wiadomość do wyświetlenia.

sendMessage(String message) – wysyła wiadomość z ostrzeżeniem za pomocą API usługi CallMeBot. Argument to treść wiadomości do wysłania.

handleRoot() – obsługuje żądania HTTP do strony głównej serwera. Wyświetla formularz konfiguracji sieci i numeru telefonu.

handleSave() –obsługuje zapis konfiguracji (SSID, hasło Wi-Fi, numer telefonu, klucz API) przesłanej przez formularz

resetConfig() –czyści zapisane dane konfiguracyjne w pamięci nieulotnej Preferences i resetuje ESP32

6. Specyfikacja zewnętrzna urządzenia

6.1 Opis funkcji elementów sterujących urządzeniem

Projekt jest sterowany przez trzy przyciski

- Przycisk do zresetowania ustawień Access Pointa – hasła i nazwy sieci WiFi do której ma zostać przyłączony, a także numer telefonu i kod dostępu do API generowany automatycznie w aplikacji WhatsApp
- Przycisk służący do wyłączenia alarmu, który sam wykrywa czy nieprawidłowość została zgłoszona przez MQ – 135 czy przez czujnik ruchu PIR. W przypadku wykrycia intruza automatycznie wyłącza komponent
- Przycisk odpowiedzialny za włączenie lub wyłączenie czujnika PIR. Zastosowano tutaj opóźnienie – wiadomość i zmianie stanu elementu zostanie pokazana na wyświetlaczu od razu, jednakże sam czujnik włączy/wyłączy się dopiero po minucie.

6.2 Opis elementów wykonawczych

Urządzenie swoją funkcjonalność i przejrzystość zawdzięcza wyświetlaczowi OLED, który informuje użytkownika o zebranych pomiarach, o stanie alarmu i czujnika PIR, a także o stanie połączenia z siecią WiFi. Informowanie użytkownika o włączonym lub wyłączonym alarmie zachodzi również dzięki czerwonej diodzie LED, która podczas uruchomionego alarmu jest włączona, a podczas wyłączonego alarmu jest zgaszona i zapala się zielona dioda. Samo informowanie o alarmie odbywa się również poprzez głośny Buzzer, który uruchamia się gdy zostało wykryte zagrożenie.

6.3 Opis reakcji oprogramowania na zdarzenia zewnętrzne

Za każdym razem gdy użytkownik użyje danego przycisku przesyłana jest informacja do mikrokontrolera i podejmowane są kroki w tym zmiana tekstu lub uruchomienie alarmu. Uruchamia się on gdy zaistnieją dane sytuacje: w powietrzu zostanie wykryty łatwopalny gaz lub wykryty zostanie intruz. Układ wyśle wtedy powiadomienie do użytkownika. Dodatkowo wiadomość zostanie dostarczona kiedy czujnik DHT11 wykryje temperaturę lub wilgotność przekraczające odpowiednio 30 °C lub wilgotność powyżej 70%.

6.4 Instrukcja obsługi urządzenia

- Podłącz układ do Powerbanka
- Sprawdź, czy układ działa poprawnie oraz czy wyświetlane są informacje na wyświetlaczu
- Pobierz aplikację WhatsApp i dodaj na niej do swoich kontaktów *CallMeBot API*.
- Wyślij wiadomość w celu uzyskania kodu dostępu:
„I allow callmebot to send me messages”
- Połącz się z siecią Wi-Fi *ESP32_Config*
- Wejdź na stronę pod adresem *192.168.4.1*, wpisz wymagane dane i zatwierdź. Mikrokontroler powinien się zresetować i wyświetlić na wyświetlaczu informację i połączeniu. W przypadku wpisania złych danych lub chęci zresetowania ustawień należy kliknąć górny przycisk
- Włącz czujnik ruchu za pomocą dolnego przycisku – czujnik powinien aktywować się po upływie jednej minuty. Ten sam przycisk służy też do wyłączenia go
- Sprawdź, czy po włączeniu alarmu otrzymasz odpowiednią wiadomość SMS. Alarm wyłączany jest środkowym przyciskiem

6.5 Opis sposobu programowania układu

Testy polegały na sprawdzeniu poprawności działania każdego z czujników oraz czy wyświetlanie tekstu zmienia się poprawnie, a także czy przyciski reagują na kliknięcia i nie występują przekłamania. Testowanie polegało również na dotykaniu czujników i sprawdzaniu zmiany temperatury. Sam dotyk pomógł również wykluczyć ewentualne zwarcia i przegrzewanie się pojedynczych elementów układu. Czujnik tlenku węgla i łatwopalnych gazów testowano poprzez użycie zapalniczki i podawanie strumienia gazu z niej ulatniającego się. Czujnik ruchu był testowany na zasadzie machania przed nim. Wszystkie testy przebiegły pomyślnie i spełniły założenia projektu określone w karcie.

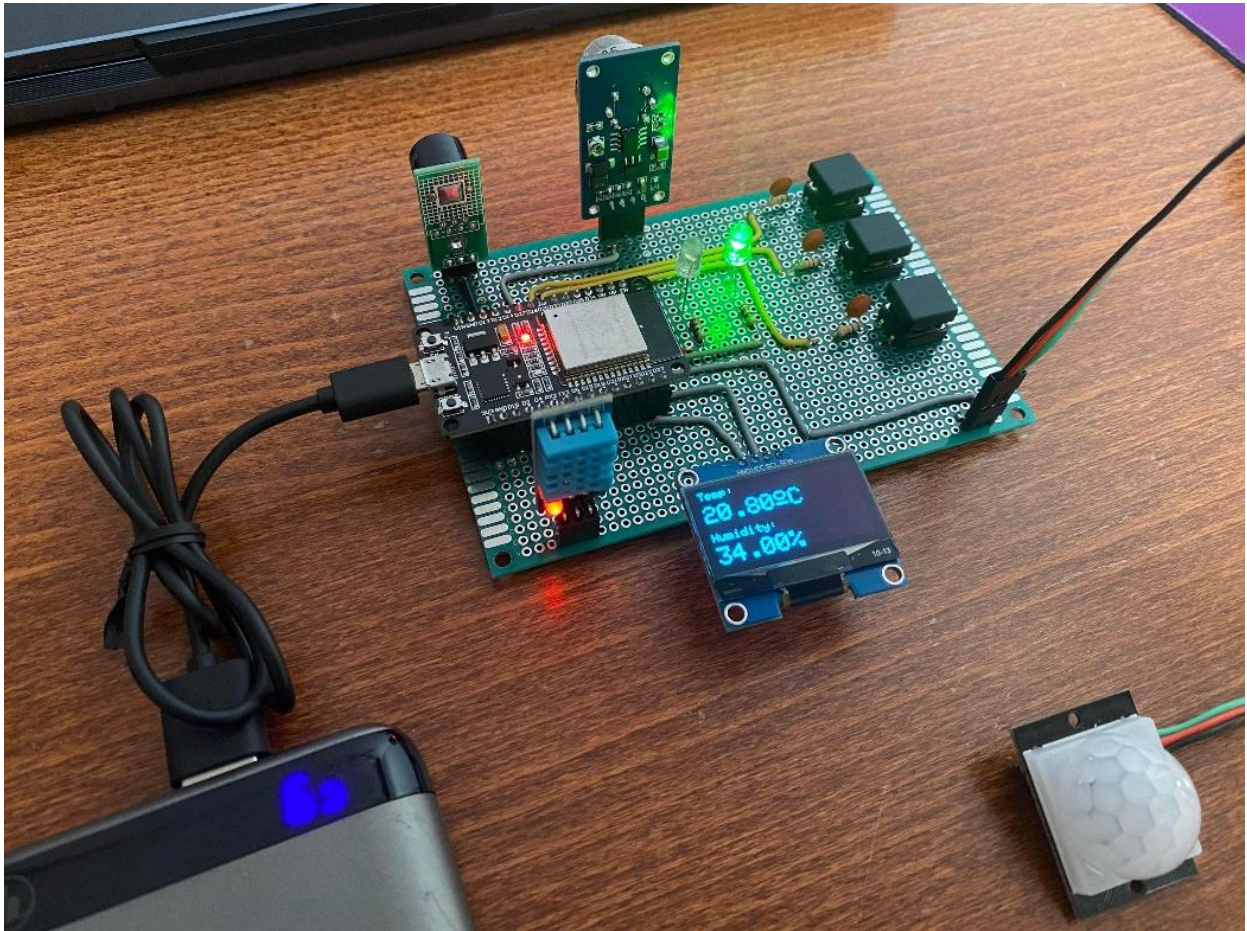
7. Wnioski i uwagi z przebiegu pracy

Problemy jakie wynikły podczas pracy nad układem w głównej mierze były skutkiem błędnie napisanego kodu, lub ze złego połączenia elementów na płytce stykowej. Zostały one naprawione poprzez dokładne przejrzanie dokumentacji i natychmiastowe wprowadzenie poprawek w napisanym kodzie jak i w montażu, dzięki czemu układ był testowany na bieżąco po każdej poprawce. Proces lutowania przebiegł pomyślnie, mimo braku wcześniejszej wiedzy w tym zakresie. Wszelkie połączenia były od razu testowane multimetrem w celu wykluczenia ewentualnego zwarcia.

8. Wnioski końcowe

Projekt był ciekawym doświadczeniem, dzięki któremu zainteresowano się mikroelektroniką. Prototypowanie na uniwersalnej płytce pozwoliło na szybkie testowanie i udoskonalanie projektu. Montaż na uniwersalnej płytce lutowniczej umożliwił zdobycie praktycznych umiejętności w tworzeniu układów elektronicznych. Temat projektu pokazał jak duże możliwości daje integracja pojedynczych czujników. Natomiast realizacja systemu z alarmem, wyświetlaczem i powiadomieniami SMS poszerzyła wiedzę z zakresu programowania i komunikacji urządzeń.

9. Zdjęcie prezentujące gotowy projekt



10. Literatura

- ESP32: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf
- MQ – 135: <https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf>
- PIR Sensor: <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>
- OLED SSD1306: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>
- DHT11: <https://www.tme.eu/Document/7a4fd48d400b8c4c8309ef1e2b13cdd4/MR003-005-1.pdf>

