

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI, FOTONIKI I
MIKROSYSTEMÓW

KIERUNEK: ELEKTRONIKA (EKA)

SPECJALNOŚĆ: APARATURA ELEKTRONICZNA (EAE)

PRACA DYPLOMOWA
INŻYNIERSKA

Systemy automatyki i monitoringu
w inteligentnym budynku

Automation and monitoring systems
in a smart building

AUTOR:

Mateusz Radosław Graczyk

PROWADZĄCY PRACĘ:

Dr inż. Mariusz Ostrowski, W12N

OCENA PRACY:

WROCŁAW, 2022

Spis treści

1.	Wstęp.....	4
1.1.	Cel i zakres pracy	4
1.2.	Założenia projektowe	5
2.	Inteligentne systemy dostępne na rynku	6
3.	Zastosowane elementy elektroniczne	10
3.1.	Arduino Uno oraz Arduino Nano	10
3.2.	Serwomechanizm Feetech FS90R	11
3.3.	Czujnik PIR HC-SR501.....	12
3.4.	Moduł Bluetooth HC-05	13
3.5.	Adresowalne diody LED WS2812B.....	15
3.6.	Układ scalony MSGEQ7	16
3.7.	Wentylator DFR0332	19
3.8.	Przetwornica impulsowa step-down (12V/5A do 5V/5A)	20
4.	Budowa urządzeń	21
4.1.	Inteligentny budynek	21
4.1.1.	Schemat blokowy	21
4.1.2.	Schemat połączeń systemu.....	22
4.1.3.	Montaż elementów	24
4.2.	Wyświetlacz widma dźwięku	26
4.2.1.	Schemat blokowy	26
4.2.2.	Schemat połączeń elementów	28
4.2.3.	Montaż elementów	30
5.	Oprogramowanie	32
5.1.	Inteligentny budynek	33
5.1.1.	Konfiguracja.....	33

5.1.2.	Główny algorytm.....	34
5.1.3.	Kluczowe funkcje.....	36
5.1.4.	Aplikacja mobilna	38
5.2.	Wyświetlacz widma dźwięku	41
5.2.1.	Konfiguracja.....	41
5.2.2.	Główny algorytm.....	43
5.2.3.	Kluczowe funkcje.....	44
5.2.4.	Aplikacja mobilna	46
6.	Testy	49
6.1.	Inteligentny budynek	49
6.2.	Wyświetlacz widma dźwięku	50
7.	Podsumowanie i wnioski.....	52
	Spis ilustracji	53
	Spis tabel	54
	Bibliografia.....	54

1. Wstęp

Od zarania dziejów człowiek dąży do poprawy swojego komfortu i poczucia bezpieczeństwa. Przyspieszenie procesu budowy lepszego życia na Ziemi w dużej mierze zawdzięczamy rozwojowi nauki i technologii. Dokonane odkrycia, pozwoliły na modernizację pospolitych budynków, wskutek czego codzienne funkcjonowanie stało się przyjemne i proste. Nic dziwnego, że zainteresowanie inteligentnymi budynkami z roku na roku wzrasta. Istnieje ku temu wiele powodów. Skuteczna automatyzacja rutynowych zadań, czy zmniejszenie kosztów eksploatacji to główne zalety, dzięki którym przybywa użytkowników inteligentnych systemów. Produkcji, konkurując ze sobą, oferują coraz lepsze rozwiązania w przystępniejszych cenach.

Rosnąca popularność szerokorozumianych inteligentnych budynków zaowocowała chęcią zagłębienia się w dany temat. W niniejszej pracy przedstawiono realizację projektu inteligentnego budynku oraz wyświetlacza widma dźwięku, który mógłby być częścią jego wyposażenia.

Opracowano rozbudowany system zarządzania inteligentnym budynkiem, stawiając na praktyczne rozwiązania, zapewniające bezpieczeństwo i komfort użytkowania.

Wyświetlacz widma dźwięku został zaprojektowany w celu dodania efektów wizualnych do odtwarzanej muzyki, zapewniając tym samym niesamowite wrażenia podczas słuchania ulubionych utworów.

Za sterowanie inteligentnym budynkiem oraz wyświetlaczem widma dźwięku odpowiadają dwie, niezależnie działające aplikacje mobilne, które stworzono na urządzenia z systemem operacyjnym Android.

1.1. Cel i zakres pracy

Celem pracy jest opracowanie i stworzenie budynku z rozbudowanym systemem automatyki i monitoringu, oraz wyświetlacz widma dźwięku będącego częścią jego wyposażenia. Oba projekty sterowane są za pomocą osobnych aplikacji mobilnych na urządzenia z systemem Android.

Następujące elementy stanowią zakres pracy:

- badanie rynku inteligentnych budynków,
- zebranie wymagań funkcjonalnych i konstrukcyjnych,
- opracowanie koncepcji systemu oraz wyświetlacza,
- implementacja rozwiązań,
- testowanie funkcjonalności systemu i wyświetlacza.

Dodanie do budynku systemów automatyki i monitoringu, uczyni go inteligentnym oraz zwiększy bezpieczeństwo i komfort życia domowników.

1.2. Założenia projektowe

Projekt zakłada zaprojektowanie i wykonanie dwóch urządzeń, działających na osobnych modułach Arduino oraz sterowanych za pomocą oddzielnych aplikacji mobilnych.

Pierwszym zadaniem jest budowa konstrukcji oraz systemu inteligentnego budynku. Drugim, natomiast wyświetlacza widma dźwięku. Rozdzielenie urządzeń jest konieczne ze względu na rozmiar wyświetlacza.

W ramach wymagań funkcjonalnych inteligentnego budynku określono:

- oświetlenie z możliwością regulacji natężenia światła,
- czujnik z sygnalizacją wykrycia ruchu,
- wentylacja z możliwością regulacji obrotów,
- otwieranie i zamykanie bramy garażowej,
- sterowanie za pomocą aplikacji mobilnej na urządzenia z systemem Android.

Następnie dokonano spis wymagań konstrukcyjnych oraz sprzętowych:

- budowa makiety budynku,
- podłączenie elementów elektronicznych systemu,
- oprogramowanie Arduino Uno za pomocą języka programowania C++ oraz środowiska programistycznego Arduino IDE.

Wymagania funkcjonalne wyświetlacza widma dźwięku:

- efekt wizualny z wykorzystaniem diod LED WS2812B,
- ilość zapalanych diod LED zależnie od amplitudy poszczególnych zakresów częstotliwości (63Hz, 160Hz, 400Hz, 1kHz, 2.5kHz, 6.25kHz oraz 16kHz),
- regulacja barwy,
- sterowanie za pomocą aplikacji mobilnej na urządzenia z systemem Android.

W wymaganiach konstrukcyjnych wyświetlacza widma dźwięku wyróżnia się:

- Urządzenie wykorzystujące Arduino Nano z mikrokontrolerem ATmega328 AVR,
- Oprogramowanie stworzone w środowisku Arduino IDE z wykorzystaniem języka programowania C++,
- Analiza amplitud częstotliwości za pomocą układu scalonego MSGEQ7.

2. Inteligentne systemy dostępne na rynku

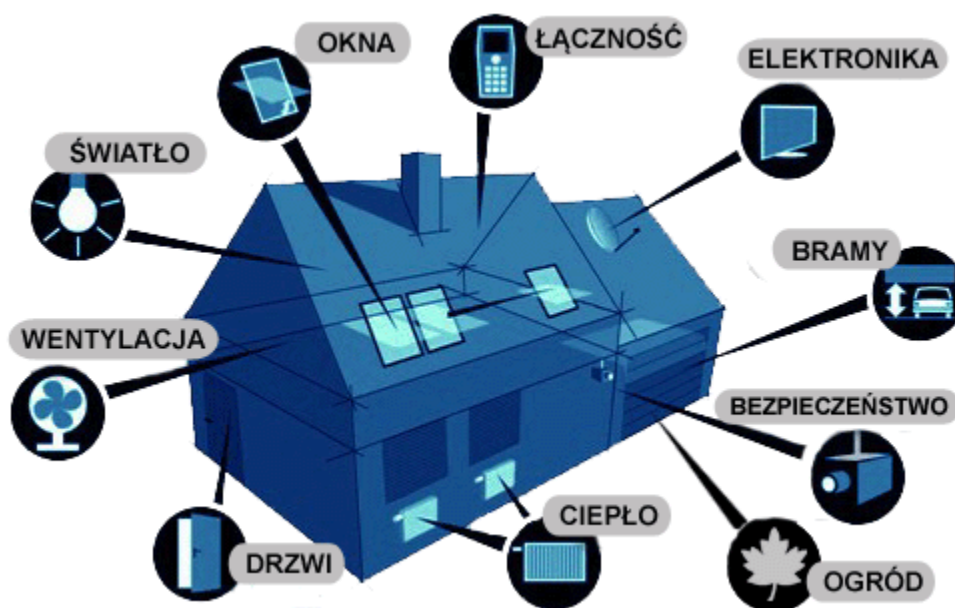
Rynek inteligentnych budynków intensywnie się rozwija. Osoby prywatne jak i przedsiębiorstwa coraz częściej korzystają z udogodnień, które zapewniają im producenci inteligentnych rozwiązań. Firmy, rywalizując ze sobą o potencjalnych klientów, prezentują produkty o lepszych parametrach i przystępniejszych cenach. Z roku na roku ilość zadowolonych użytkowników rośnie.

Systemy dostępne na rynku, tworzone przez wielkie korporacje, mogą oferować bardzo zaawansowane funkcjonalności, na przykład: zbierając informacje na temat sposobu funkcjonowania użytkownika, mogą dostosowywać swoje parametry zapewniając lepsze właściwości działania systemu. Przytoczyć można tu głowice termostatyczną marki Fibaro, która wykorzystując nowatorskie algorytmy, początkowo „uczy” się danego pomieszczenia. Precyzyjnie sprawdza kubaturę wnętrza, szybkość reakcji na podnoszoną temperaturę oraz czas ochładzania. Na podstawie otrzymanych wyników ustala preferencje przestrzeni i koryguje sprzęt, co przekłada się to na podnoszenie komfortu cieplnego oraz oszczędności w perspektywie czasu. Redukcja kosztów według badań firmy wynosi nawet 42% [1].

Na zaawansowanie systemu składają się nie tylko możliwości poszczególnych urządzeń, ale również ich ilość. Wiadomym jest, że im więcej inteligentnego wyposażenia tym korzystniej dla użytkowników. Nie ma jednoznacznej granicy, kiedy zwykła automatyka danej nieruchomości, przekształca ją w pełnoprawny inteligentny budynek. Jednakże stosunkowo często „inteligencja” w elektronice, oznacza spełnienie kilku poniższych warunków:

- zdolność adaptacji do zmiennych warunków,
- możliwość modyfikacji zdalnej,
- wstępną obróbkę zmierzonych danych,
- możliwość podejmowania decyzji w zależności od uzyskanych pomiarów,
- co najmniej kilka urządzeń współpracujących ze sobą.

Na rysunku 1. przedstawiono podstawowe komponenty budynku, które najczęściej wymagają automatyzacji.



Rys. 1 Komponenty budynku wymagające automatyzacji [2].

Na cenę inteligentnego budynku składa się wiele elementów. Najważniejszymi czynnikami są powierzchnia nieruchomości oraz stopień zaawansowanie systemu. Zależnie od indywidualności projektu ceny mogą się znacząco wahać.

Szacuje się, że koszt budowy średniozaawansowanego, przewodowego systemu, dla nieruchomości o powierzchni około 100 metrów, oscyluje w przedziale 20-30 tysięcy złotych. Najtańsze systemy zaczynają się od 10 000 złotych. Najdroższe, natomiast potrafią osiągnąć kwoty powyżej stu tysięcy złotych. Przyjmuje się jednak, że średnia minimalna stawka za podstawowy system to 200 złotych za metr kwadratowy [3].

Skład ceny zawiera instalację przewodową systemu oraz centralę, która działa jak lokalny serwer. System fizyczny (przewodowy) jest znacznie lepszy pod względem optymalizacji konkretnego budynku. Jest mniej awaryjny, niezawodny oraz szybszy niż system polegający na sieci bezprzewodowej. Warto zainteresować się systemami fizycznymi już w trakcie projektu budynku. Jeżeli klient pragnie zmodernizować już istniejącą nieruchomość, cena również wzrośnie ze względu na wymagany remont oraz zwiększony stopień skomplikowania.

W przypadku istniejących budynków, lepszym rozwiązaniem jest zastosowanie systemu bezprzewodowego, który nie ingeruje w konstrukcję budynku. Ze względu na brak okablowania koszty są znacznie mniejsze. Za rozbudowany system należy zapłacić pomiędzy 8-10 tysięcy złotych, zależnie od ilości użytych elementów takich jak: centrala, inteligentne gniazdka, oświetlenie, kamery monitoringu itp.

Ważnym atutem tego typu systemów jest możliwość stopniowej rozbudowy. W zależności od rodzajów transmisji danych i użytych technologii ceny mogą się różnić, na przykład: Firma Fibaro oferuje klientom sterowanie inteligentnym budynkiem z poziomu aplikacji, która działa w chmurze producenta. Koszt takiego rozwiązania to około dwa tysiące złotych [4].

Znacznie tańszym rozwiązaniem są pojedyncze inteligentne urządzenia lub zestawy startowe. Za pakiet zawierający: centralę systemu, kontaktron, inteligentne gniazdko oraz czujniki ruchu, dymu i zalania, w cenie około 2 tys. złotych [5].

Zazwyczaj, jednak zwykli klienci nie potrzebują zaawansowanych systemów, które mogą wymagać przestrzegania ścisłych reguł co do konstrukcji. Jest to jeden z powodów, który objawia się brakiem zainteresowania. Potencjalnych nabywców zniechęcać może, zarówno podatność elektroniki na uszkodzenia lub też względy finansowe.

Producenci wychodzą, również tutaj z szerokim wyborem rozwiązań z różnych półek cenowych. Oferują możliwość zakupów urządzeń elektronicznych, które nie ingerują w strukturę budynku oraz są znacznie tańsze od pełnych systemów automatyki. Przykładem mogą być inteligentne gniazdko, systemy asystenta głosowego, czy też kamery monitoringu.



Rys. 2 Inteligentne gniazdko WiFi Gosung SP1 [6].

Inteligentne gniazdko, zaraz po wszelkiego rodzaju inteligentnych czujnikach są najczęściej wybieranym gadżetem, dzięki którym kupujący rozpoczynają swoją przygodę z „Smart Home”. Gniazdko umożliwia sterowanie się za pomocą smartfonu podłączonego do sieci WiFi

lub poprzez komunikację Bluetooth. Niekiedy wybrane modele mogą, również współpracować z asystentami głosowymi, takimi jak: Siri, Amazon Alexa lub Asystent Google.

Oferują ustalenie harmonogramów włączenia i wyłączenia zasilania. Ponadto mogą udostępniać statystyki na temat zużycia energii. Są w pełni bezpiecznym urządzeniem, ze względu na posiadane zabezpieczenia przed przepięciem.

Kolejnym ciekawym rozwiązaniem na rynku są zestawy startowe do automatyki budynku, które cieszą się, również niemałą popularnością. Jest to kilka podstawowych urządzeń połączonych w pakiet, pozwalający zmodernizować budynek, w bardzo prosty sposób. Najczęściej na zestaw składają się pojedyncze urządzenia, które zaprogramowano tak, aby współpracowały ze sobą. Dzięki ich połączeniu w zestaw, ceny są korzystniejsze dla kupujących.



Rys. 3 Zestaw sześciu inteligentnych urządzeń marki Fibaro [7].

Zestaw przedstawiony na rysunku 3. można kupić w sklepie internetowym za kwotę około 2 200zł. Koszt tych samych urządzeń, kupionych, na tej samej stronie, zaprezentowano poniżej:

- centrala systemu „Fibaro Home Center Lite”: 1 225zł/szt. [8],
- Czujnik ruchu, światła, temperatury „Fibaro Motion Sensor”: 250zł/szt. [9],
- Czujnik dymu „Fibaro Smoke Sensor”: 280zł/szt. [10],
- Czujnik zalania „Fibaro Flood Sensor”: 260zł/szt. [11],
- Czujnik zbliżeniowy z pomiarem temperatury „Fibaro Door Window Sensor”: 270zł/szt. [12],
- Inteligentne gniazdko z bolcem „Fibaro Wall Plug”: 280zł/szt. [13].

Suma wyżej wymienionych urządzeń, przekłada się na kwotę 2 565 złotych, co daje łącznie 365 złotych więcej, niż w przypadku kupna zestawu.

3. Zastosowane elementy elektroniczne

Niniejszy rozdział został poświęcony przedstawieniu i omówieniu zastosowanych elementów elektronicznych. Szczególną uwagę zwrócono na dane techniczne, które wpłynęły na wybór części, składających się na pełny projekt. Głównym kryterium była minimalizacja kosztów oraz umożliwienie budowy urządzeń zgodnie z ustalonymi założeniami. W tym celu część podzespołów wykorzystano z innych projektów realizowanych w trakcie toku studiów.

3.1. Arduino Uno oraz Arduino Nano

Logiką systemu inteligentnego budynku będzie zarządzać Arduino Uno R3 (rys. 4.). Podany model wyposażony jest w 8-bitowy mikrokontroler ATmega328P z serii megaAVR. Posiada 14 uniwersalnych pinów cyfrowych wejścia lub wyjścia. Sześć z nich można wykorzystać jako wyjścia PWM (ang. Pulse-Width Modulation) – modulacja szerokości impulsu. Dodatkowo moduł został wyposażony w 6 wejść analogowych [14] [15].

Producent umożliwia podłączenie zasilania poprzez:

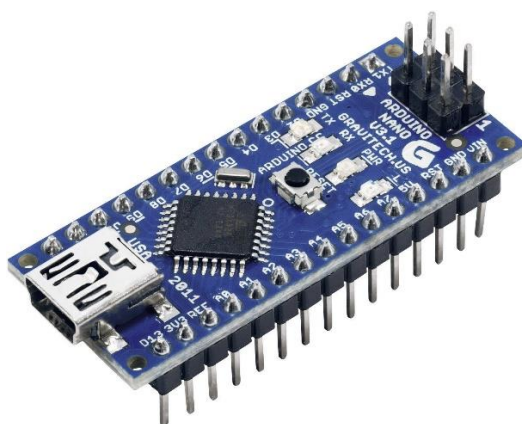
- gniazdo USB typu A-B,
- gniazdo zasilania,
- pin Vin na module [14].

Zaleca, również stosowanie napięcia DC z zakresu 7-12V. Przekroczenie napięcia 20V, może spowodować uszkodzenie układów, z kolei poniżej 6V, niestabilną pracę lub też jej brak [14] [15].



Rys. 4 Ilustracja pogładowa modułu Arduino Uno [16].

Wyświetlacz widma dźwięku sterowany będzie za pomocą modułu Arduino Nano (rys. 5.). Zarówno ten model, jak i wcześniej wymieniony Arduino Uno, posiadają jednakowy mikrokontroler oraz większość parametrów technicznych. Najbardziej widoczną różnicą jest rozmiar urządzenia. Jej wymiary to zaledwie: $45 \times 18\text{mm}$. Kolejnym zauważalnym odstępstwem jest zastosowanie złącza mini USB, zamiast USB typu B oraz większą ilość pinów analogowych [17].



Rys. 5 Ilustracja pogładowa modułu Arduino Nano [18].

3.2. Serwomechanizm Feetech FS90R

Otwieranie i zamykanie bramy garażowej modelu inteligentnego budynku, umożliwi serwomechanizm Feetech FS90R. Wybrany model pozwala na nieprzerwaną rotację o kąt 360° . Serwomechanizm wymaga napięcia zasilania wynoszącego 3.3V lub 5V, przy poborze prądu około 10mA w spoczynku oraz 120mA podczas pracy bez obciążenia.



Rys. 6 Ilustracja poglądowa mikro serwomechanizmu SG90 [19].

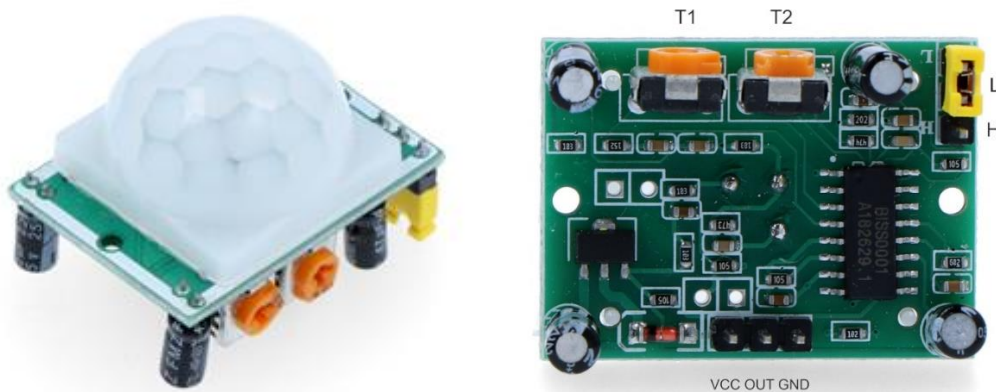
Dużymi zaletami wybranego komponentu są:

- niewielka waga wynosząca 9 gram,
- mały rozmiar: $23.2 \times 12.5 \times 22\text{mm}$,
- moment: 1,5kg/cm przy 4.8V [20].

Sterowanie odbywa się za pomocą sygnału PWM. Dla wypełnienia 1.5ms serwomechanizm pozostaje w spoczynku. Wartość tą można regulować ustawiając odpowiednio potencjometr, znajdujący się na spodzie obudowy. Zmniejszenie szerokości impulsu spowoduje ruch w kierunku zgodnym z ruchem wskazówek zegara, przy czym prędkość wzrośnie wraz ze zmniejszeniem wymienionej wartości. W sytuacji zwiększenia szerokości impulsu, serwomechanizm zacznie się obracać w kierunku przeciwnym do ruchu wskazówek zegara [20].

3.3. Czujnik PIR HC-SR501

Czujnik elektroniczny PIR (ang. Passive Infra Red) – model HC-SR501, jest to pasywny czujnik podczerwieni, pozwalający na wykrycie ruchu. Czujniki tego typu znajdują zastosowanie w systemach alarmowych, systemach automatycznego włączania oświetlenia itp. Głównym modułem jest detektor podczerwieni tzw. Pyroelement, który umiejscowiony jest na środku modułu pod plastikową soczewką Fresnela. Zastosowanie jej pozwala na skupianie padającego przez nie światła na detektorze [21].



Rys. 7 Ilustracja pogładowa czujnika PIR HC-SR501 [22].

Wykrywanie ruchu polega na bardzo precyzyjnym pomiarze temperatury. Każda zmiana, na wyższą lub niższą, jest traktowana jako alarm. Zakres napięcia zasilania modułu wynosi pomiędzy 4.5V a 20V. Kąt widzenia czujnika to około 100°, a maksymalny zasięg to 7m. Stopień czułości (pole, w którym detektor wykrywa ruch) czujnika, ustawia się potencjometrem T1, natomiast potencjometr T2 odpowiedzialny jest za czas trwania stanu wysokiego po wykryciu ruchu (rys. 7) [22]. Piny VCC oraz GND, służą do podłączenia zasilania. Przez pin OUT wysyłany jest sygnał w momencie wykrycia ruchu. Zworka znajdująca się na module w pozycji „L” lub „H” ustawia jeden z dwóch trybów pracy czujnika:

- tryb powtarzalny (H) – na pinie OUT zostanie ustawiony stan wysoki (3.3V), w momencie wykrycia. Wartość ta zostanie obniżona po określonym czasie zależnie od potencjometru T2,
- tryb niepowtarzalny (I) – na pinie OUT zostanie ustawiony stan wysoki (3.3V), w momencie wykrycia. Wartość będzie utrzymywana, aż do momentu opuszczenia przez wykryty obiekt danego obszaru [23].

3.4. Moduł Bluetooth HC-05

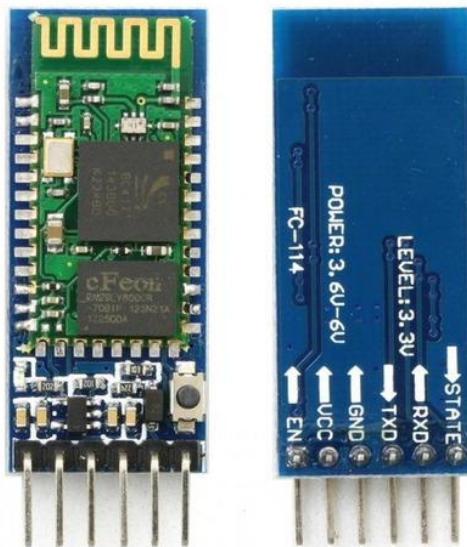
Projekt wymaga opracowania komunikacji pomiędzy inteligentnym budynkiem, wyświetlaczem widma dźwięku, a smartfonem z systemem operacyjnym Android. Istnieje kilka rozwiązań potrafiących zapewnić bezprzewodowe połączenie urządzeń.

Poniżej przedstawiono kilka standardów bezprzewodowej transmisji danych:

- NRF,

- ZigBee,
- Wi-Fi,
- Bluetooth [24].

W celu zapewnienia połączenia wybrano moduł Bluetooth HC-05. Urządzenie pozwala na bezprzewodowe przesyłanie danych na odległość maksymalnie 10 metrów. W module zastosowano technologie Bluetooth 2.0 pracującą z częstotliwością 2.4GHz, z tego powodu uzyskano kompatybilność z większością urządzeń. Z kolei prędkość transferu danych może wynieść nawet 1Mbps. Moduł zasilany jest napięciem w zakresie 4 – 6V, a pobór prądu w czasie parowania wynosi 30mA, natomiast w trakcie samej wymiany danych maleje do 8mA [25] [26].



Rys. 8 Ilustracja poglądowa modułu Bluetooth HC-05 [27].

Bluetooth HC-05 wyposażony został w 6 pinów:

- VCC – zasilanie modułu,
- GND – masa,
- TXD – transmisja danych,
- RXD – odbieranie danych,
- STATE – używany do sprawdzania, czy urządzenie jest podłączone, czy też nie,
- EN – służy do przełączenia modułu w stan komend AT, służącego do konfiguracji modułu, na przykład: ustawienie nazwy urządzenia lub hasła [28].

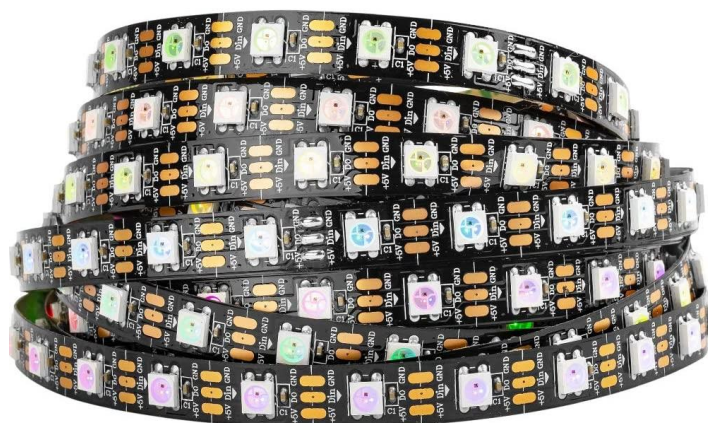
3.5. Adresowalne diody LED WS2812B

Adresowalne diody LED, wykorzystywane są w urządzeniach, gdzie wymagany jest określony kolor (jasność) odpowiednich diod. Przykładem mogą być telefony z ekranem AMOLED lub też tablice informacyjne znajdujące się na przystankach w większych miastach. Zastosowanie tego typu technologii wiąże się z potrzebą umiejętnego kontrolowania dostarczanych wartości.

Na rynku istnieje wiele sterowników adresowalnych diod LED. Z serii „WS” można wymienić modele:

- WS2801 – sterownik umiejscowiony na zewnątrz obudowy oraz poprowadzona linia zegara,
- WS2811 – różniący się od poprzednika, brakiem linii zegara,
- WS2812B – model najpopularniejszy ze względu na stosunek ceny do jakości oraz oferowanych możliwości układu,
- WS2813 – jako jedyny wyposażony w zapasową linię danych, dzięki czemu w momencie przepalenia nie nastąpi zgaszenie kolejnych diod w szeregu [29] [30].

Ze względu na wystarczające parametry i możliwości funkcjonalne taśmy LED WS2812B, to właśnie ona została wykorzystana w projekcie. Dane techniczne wskazują, że podany model jest w stanie wyświetlić 16777216 kolorów z częstotliwością, nie mniejszą niż 400Hz. Wymagany zakres napięcia zasilania wynosi od 3.5V do 5.3V. Zakłada się, że poszczególna dioda RGB przy maksymalnej mocy jest w stanie zużyć 20mA prądu. W obudowie SMD 5050 umieszczone zostały diody czerwona, zielona i niebieska, dlatego można przyjąć że cała dioda WS2812B może zużyć 60mA prądu. Mając na uwadze fakt, że w projekcie jest około 200 pikseli należy dobrać zasilacz o odpowiednich parametrach [31].



Rys. 9 Ilustracja poglądowa taśmy LED WS2812B [32].

Ustawienie stanu dwóch diod LED WS2812B polega na transmisji 48 bitów, wykorzystując w tym celu jedną linię danych (DIN). Sterownik znajdujący się pierwszy w szeregu, pobierze wysłane 24 bity danych i na ich podstawie określi stopień zapalenia odpowiednich diod RGB. Na każdą diodę czerwoną, zieloną lub niebieską przypada 8 bitów danych. Jasność poszczególnych diod RGB regulujemy poprzez wysłanie wartości z zakresu 0 do 255. Maksymalne natężenie diod LED WS2812B uzyskamy dla wartości 255. Wyłączenie diody RGB nastąpi dla wartości 0. Pierwsze 8 bitów przypadnie na diodę czerwoną, kolejne na zieloną i ostatnie na niebieską. Po pobraniu przez kontroler potrzebnych 24 bitów zostaną one zredukowane, a następne dane dotrą do kolejnego kontrolera w szeregu (rys. 9.) [33].

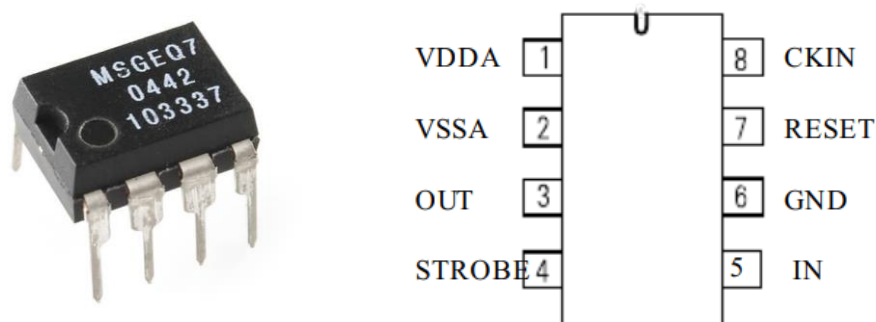


Rys. 10 Przykładowe dostarczanie danych do diod LED WS2812B [33].

3.6. Układ scalony MSGEQ7

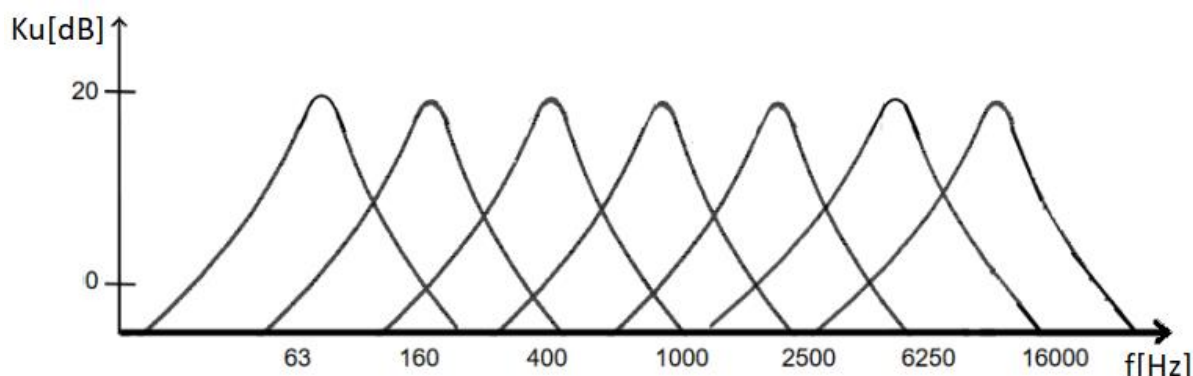
Głównym komponentem wyświetlacza widma dźwięku jest graficzny equalizer MSGEQ7 (rys. 11.). Jego funkcjonalność polega na wyznaczaniu amplitud poszczególnych częstotliwości: 63Hz, 160Hz, 400Hz, 1kHz, 2.5kHz, 6.25kHz oraz 16kHz.

Zasilic go można napięciem z zakresu 2.7V a 5.5V, natomiast przy 5V układ wskazuje najlepsze parametry pracy. Ponadto układ charakteryzuje się niskim poborem prądu (około 1mA) oraz niewielkimi rozmiarami [34].



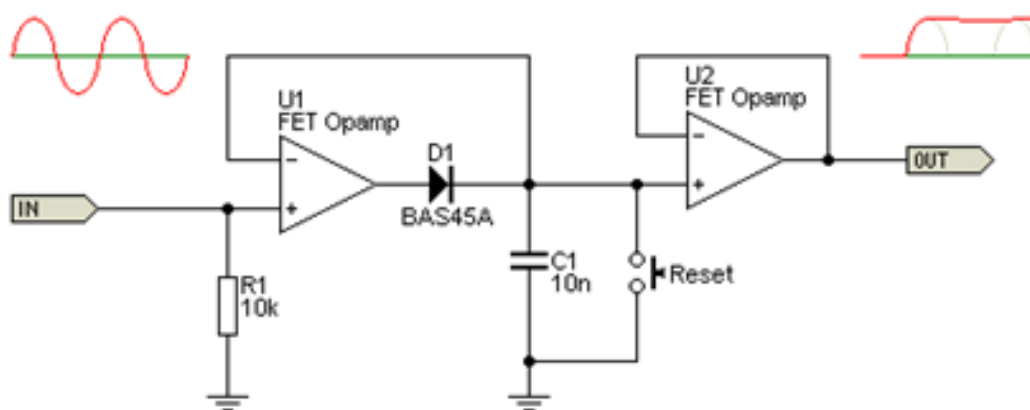
Rys. 11 Ilustracja poglądowa układu scalonego MSGEQ7 [34].

W celu detekcji szczytu odpowiednich częstotliwości, układ pobiera sygnał audio, a następnie przekazuje go do odpowiedniego filtra środkowoprzepustowego. W konsekwencji układ dokonuje przekształcenia sygnału na jeden z zakresów częstotliwości przedstawionych na rysunku 12. Zależnie od wzmocnienia wyrażanego w decybelach, wartość amplitud może się różnić [34].



Rys. 12 Przykład działania układu MSGEQ7 [34].

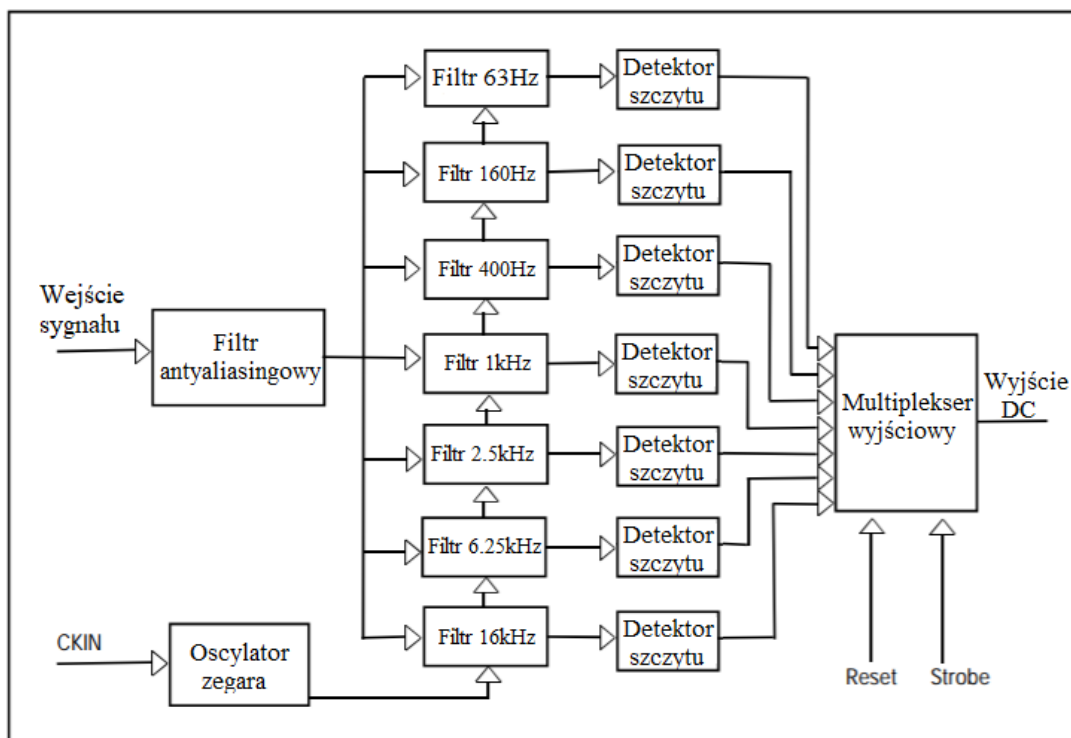
Kolejnym krokiem jest detekcja szczytu odpowiedniej częstotliwości. Przykładowe rozwiązanie znajdują się na rysunku poniżej [35].



Rys. 13 Przykładowe rozwiązanie detekcji szczytu [36].

Zagłębiając się w schemat MSGEQ7 możemy wyróżnić pięć elementów składających się na w pełni funkcjonalny układ scalony:

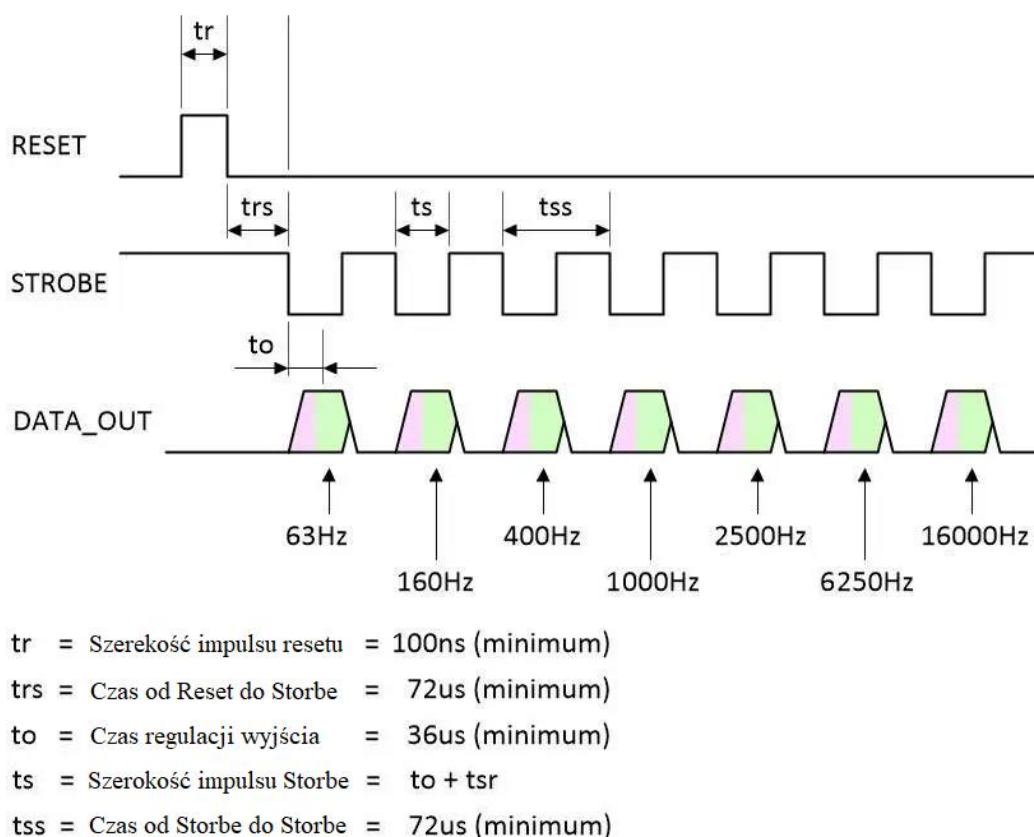
- filtr antyaliasingowy (ang. anti-alias filter) – odpowiedzialny jest za zmniejszenie zniekształceń wejściowego sygnału audio,
- filtry środkowoprzepustowe (ang. bandpass) – dzielą sygnał na siedem zakresów częstotliwości,
- oscylator zegara (ang. clock oscillator) – zegar wyzwalający rozkład sygnału dźwięku na poszczególne częstotliwości,
- detektor szczytu (ang. peak detector) – wyznacza amplitudę poszczególnych zakresów częstotliwości,
- multiplekser wyjściowy (ang. output multiplexer) – służy do wyboru jednego z kilku dostępnych sygnałów wejściowych i przekazania go na wyjście.



Rys. 14 Schemat układu MSGEQ7.

Sterowanie odbywa się za pomocą zmiany stanów na pinach Strobe oraz Reset, którymi przełączamy multiplekser wyjściowy, wskutek czego na wyjście DC podawane są wartości amplitud kolejnych zakresów. Przed odczytem danych należy zresetować multiplekser ustawiając pin Reset w stan wysoki. Następnie należy ustawić stan niski, aby wywołać linię sterującą Strobe. Kolejna zmiana napięcia linii Strobe, spowoduje uzyskanie wartości amplitudy wybranego zakresu na linii wyjścia DC. Sygnał ten może zostać odczytany za pomocą wejścia analogowego na module Arduino [37].

Wartości przedstawione na diagramie czasowym MSGEQ7, prezentują minimalny czas, jaki należy odczekać przed wykonaniem kolejnej operacji. Oznacza to, że jeżeli programista będzie zachowywał podane odstępy czasowe, nie będzie narażony na niepożądane działania.



Rys. 15 Diagram czasowy MSGEQ7.

3.7. Wentylator DFR0332

W celu prezentacji działania systemu wentylacji wykorzystano wentylator DFR0332. Jest to układ pozwalający na regulację prędkości obrotowej za pomocą sygnału PWM. Napięcie pracy układu wynosi od 3V do 6V. Przy maksymalnym napięciu zasilania bez obciążenia pozwala osiągnąć 15000obr/min. Przy tej prędkości może pobrać aż do 360mA prądu. Przy niższej prędkości obrotowej zużycie oscyluje w okolicy 40mA [38].

Poniżej przedstawiono ilustracje poglądową.



Rys. 16 Ilustracja pogładowa wentylatora DFR0332 [39].

3.8. Przetwornica impulsowa step-down (12V/5A do 5V/5A)

Przetwornica impulsowa DC-DC step-down jest to układ obniżający napięcie na wyjściu modułu. Ze względu na wymagane napięcie pracy wynoszące 5V dla wszystkich podzespołów oraz wykorzystanie zasilacza 12V/5A, zdecydowano się na zastosowanie wymienionej przetwornicy [40].



Rys. 17 Ilustracja pogładowa przetwornicy impulsowej DC-DC step-down [40].

Maksymalna moc wyjściowa przetwornic w dużej mierze zależy od sprawności przetwarzania. W przypadku wybranego modułu istotna sprawność wynosi aż 90%, a to przekłada się na maksymalną wydajność do 25W/5A. Zakres napięcia wejściowego wynosi od 12V do 24V [40].

4. Budowa urządzeń

Niniejszy rozdział prezentuje realizację fizyczną modelu inteligentnego budynku oraz wyświetlacza widma dźwięku. Skupiono się na przedstawieniu schematów działania urządzeń oraz procesie montażowym.

4.1. Inteligentny budynek

4.1.1. Schemat blokowy

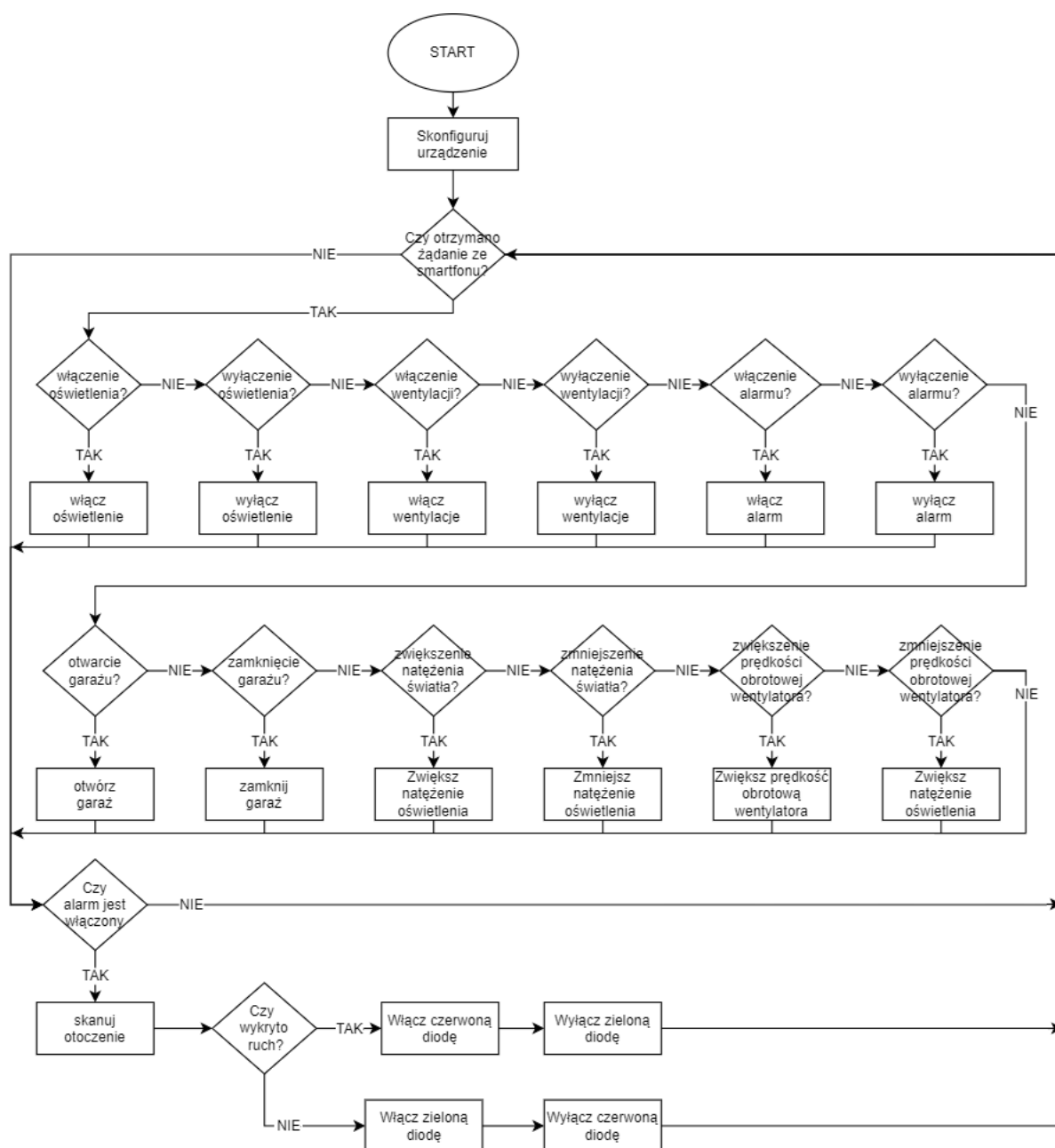
W momencie włączenia zasilania, moduł Arduino Uno konfiguruje urządzenie. Ustawiane są tryby odpowiednich pinów i następuje proces konstrukcji obiektów odpowiedzialnych za oświetlenie.

System inteligentnego budynku pracuje na zasadzie kontrolowania i obsługi żądań. W momencie otrzymania polecenia dopasowuje je do odpowiedniej funkcjonalności, a następnie je wykonuje. Możliwe są następujące rozkazy:

- włączenie lub wyłączenie oświetlenia,
- włączenie lub wyłączenie wentylacji,
- włączenie lub wyłączenie alarmu,
- otwarcie lub zamknięcie garażu,
- zwiększenie lub zmniejszenie natężenia światła,
- zmniejszenie lub zwiększenie prędkości obrotowej wentylatora.

Monitoring inteligentnego budynku składa się z trzech komponentów: czujnika PIR HC-SR501 oraz dwóch diod LED. W trakcie pracy systemu, podczas braku żądania uruchomienia alarmu, obie diody pozostają zgaszone. Stan ten utrzymywany jest do momentu otrzymania żądania z aplikacji mobilnej. Dioda zielona informuje użytkownika o włączonym trybie pracy systemu alarmowego oraz braku sygnału alarmującego. W momencie wykrycia ruchu, czujnik zawiadamia system o intruzie, wskutek czego dioda zielona zostaje zgaszona, a czerwona zapalona.

Bezpieczeństwo musi być zapewniane pomimo wysyłania innych żądań. Dlatego system inteligentnego budynku obsługuje czujnik niezależnie od otrzymania żądania, czy też nie.



Rys. 18 Inteligentny budynek – schemat blokowy systemu.

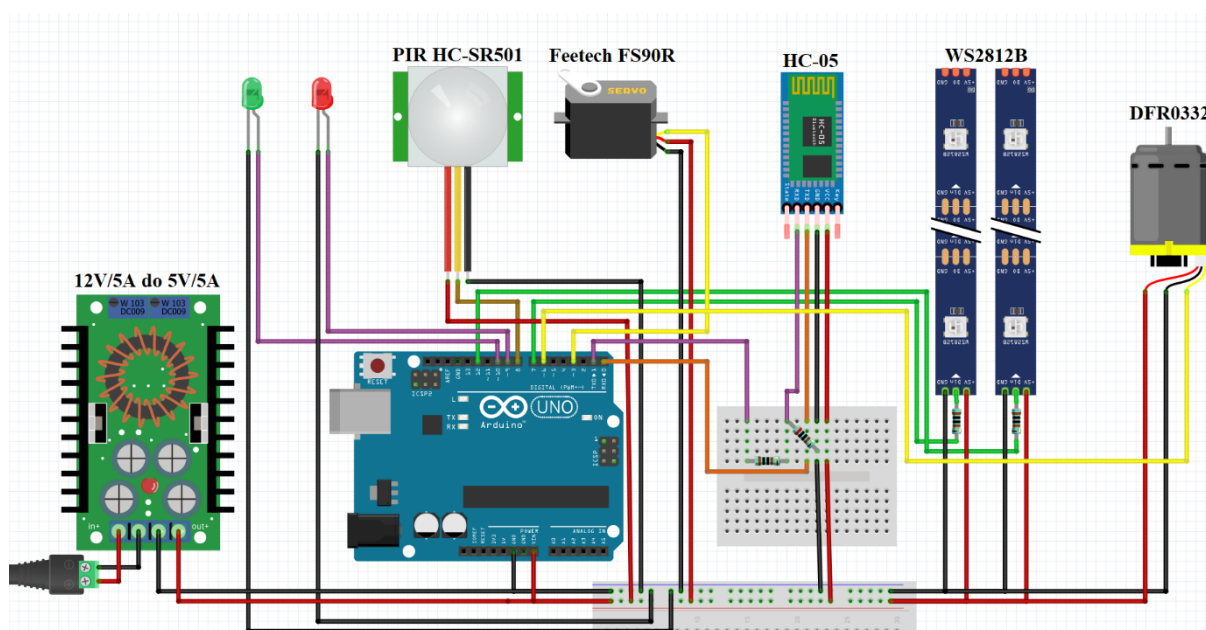
4.1.2. Schemat połączeń systemu

Zasilacz podłączono do przetwornicy impulsowej DC-DC obniżającej napięcie z 12V/5A do 5V/5A. Wyjście przetwornicy zostało podłączone do pinu Vin modułu Arduino Uno oraz prototypowej płytki stykowej. Wszystkie podzespoły systemu, za wyjątkiem diod LED sygnalizujących stan alarmu, czerpią z niej zasilanie. Wymienione diody, koloru czerwonego i zielonego, zasilane są z pinów Arduino Uno.

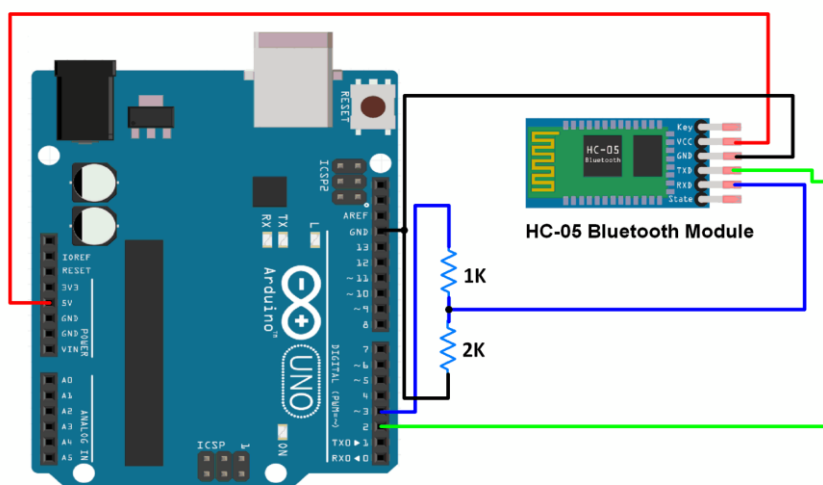
Moduł Bluetooth HC-05 wymaga zastosowania dwóch rezystorów na linii danych RX. Uzyskanie komunikacji między modułami odbywa się poprzez połączenie wejścia RX Bluetooth z pinem TX, natomiast wyjście TX z RX na Arduino Uno. Schemat podłączenia Bluetooth'a przedstawia rysunek 20.

Na początku linii danych diod LED WS2812B zastosowano rezystor $1k\Omega$ w celu ochrony pierwszej diody w szeregu przed uszkodzeniem.

Serwomechanizm Feetech FS90R oraz wentylator DFR0332 zostały podłączone do pinów umożliwiających sterowanie PWM.



Rys. 19 Inteligentny budynek – schemat połączeń systemu.



Rys. 20 Schemat podłączenia Bluetooth HC-05 z Arduino Uno [41].

4.1.3. Montaż elementów

Montaż elementów rozpoczęto od budowy makiety inteligentnego budynku. Do tego celu wykorzystano wycięty laserowo model ze sklejk. Efekt po dopasowaniu i złączeniu wszystkich elementów przedstawiono na rysunku 21.

Kolejnym etapem był montaż podzespołów inteligentnego budynku składającego się na pełny system. W tym celu wykorzystano następujące urządzenia:

- Arduino Uno – logika systemu,
- czujnik PIR HC-SR501 – monitoring bezpieczeństwa budynku,
- diody czerwona i zielona – sygnalizacja wykrycia ruchu,
- serwomechanizm Feetech FS90R – otwieranie i zamykanie bramy garażowej,
- Bluetooth HC-05 – komunikacja ze smartfonem,
- Adresowalne diody LED WS2812B – oświetlenie makiety,
- Wentylator DFR0332 – wentylacja pomieszczenia,
- Przetwornica impulsowa DC-DC step-down – obniżenie napięcia, w celu poprawnego zasilania systemu.

System zrealizowano zgodnie ze schematem połączeń przedstawionym na rysunku 19.



Rys. 21 Makieta inteligentnego budynku

Poszczególne urządzenia zostały przytwierdzone do makiety, aby uniknąć uszkodzeń w trakcie pracy. Ponadto przewody skręcono oraz miejscami skleiono, żeby wyeliminować możliwość przypadkowego rozłączenia. Starano się, również maksymalnie ograniczyć widoczność okablowania. Zasilacz umieszczono wewnątrz makiety, w celu uzyskania jednolitej konstrukcji. Zakończona praca została zaprezentowana na rysunku 22.



Rys. 22 Gotowy inteligentny budynek.

4.2. Wyświetlacz widma dźwięku

4.2.1. Schemat blokowy

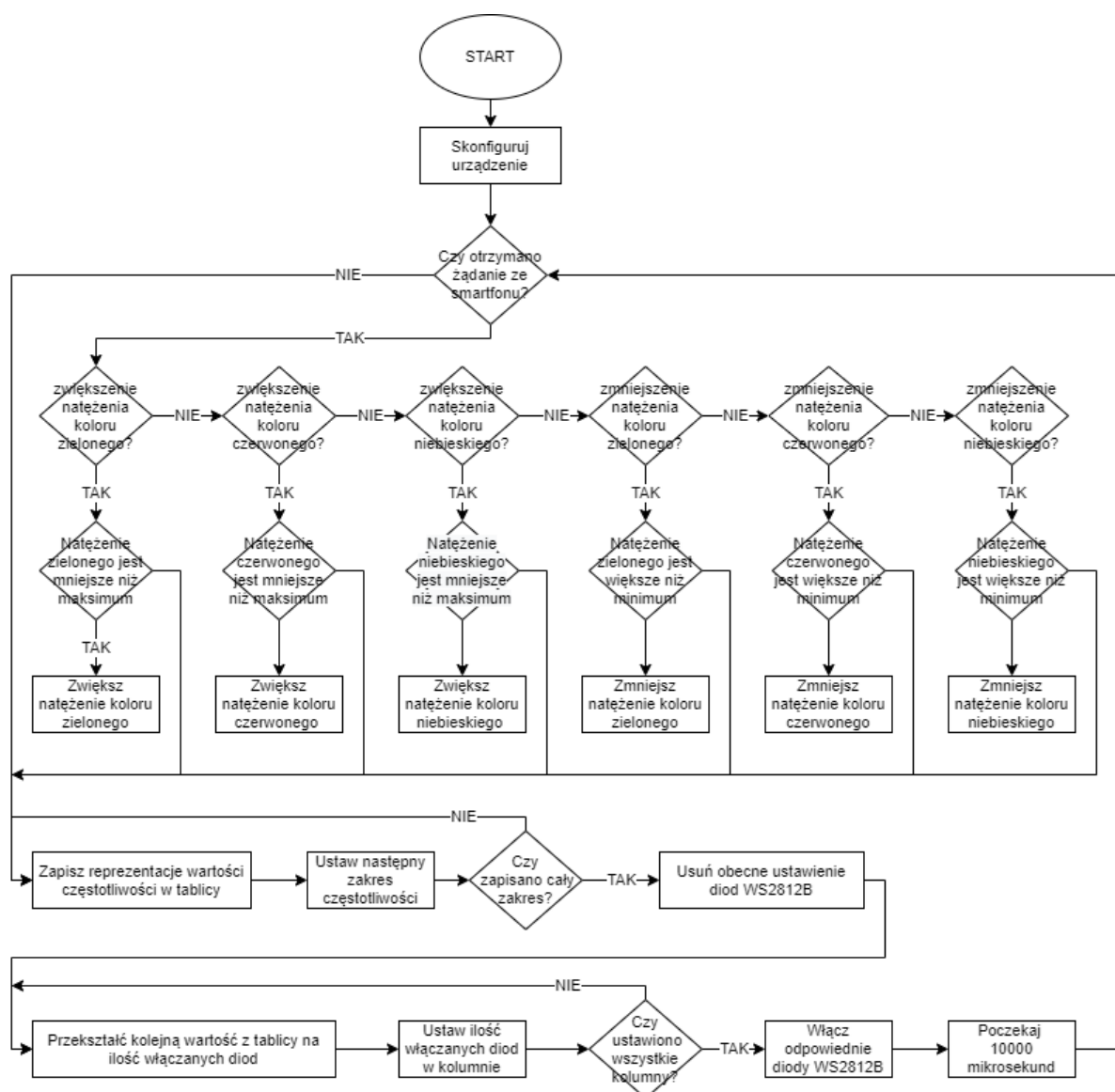
Program urządzenia rozpoczyna swą pracę w momencie podłączenia do zasilania. Na początku następuje ustawienie funkcji pinów oraz reset multiplexera będącego częścią układu MSGEQ7.

Po wykonaniu konfiguracji oprogramowanie sprawdza czy nastąpiło żądanie modyfikacji koloru adresowalnych diod LED WS2812B. Jeżeli odebrano rozkaz z aplikacji, następuje dobór odpowiedniej funkcji w celu zwiększenia lub zmniejszenia natężenia wybranego koloru. Zanim, jednak to nastąpi, oprogramowanie weryfikuje czy zmiana wartości nie przekroczy ustalonego zakresu. W wyniku tych działań natężenie może zostać zmienione lub też nie.

Niezależnie od żądania system zapisuje reprezentacje maksymalnej amplitudy ustawionego zakresu w tablicy. Następnie przestawia multiplexer układu MSGEQ7 na kolejną częstotliwość. Proces ten wykonuje się 7 razy, ponieważ wynika to z możliwości detekcji jedynie siedmiu zakresów częstotliwości.

Po wypełnieniu tablicy, następuje ich przekształcenie na wartość odpowiadającą ilości zapalanych diod LED WS2812B. Wykonanie tej czynności umożliwia ustawienie odpowiednich diod, które zostają włączone. Domyślnie ustawiany jest kolor biały, jednakże jego dobór definiowany jest na podstawie wcześniej wykonanego żądania z aplikacji mobilnej.

Przed rozpoczęciem kolejnego cyklu, program czeka 10000 mikrosekund, aby zrównoważyć prędkość zmian wyświetlanych diod.



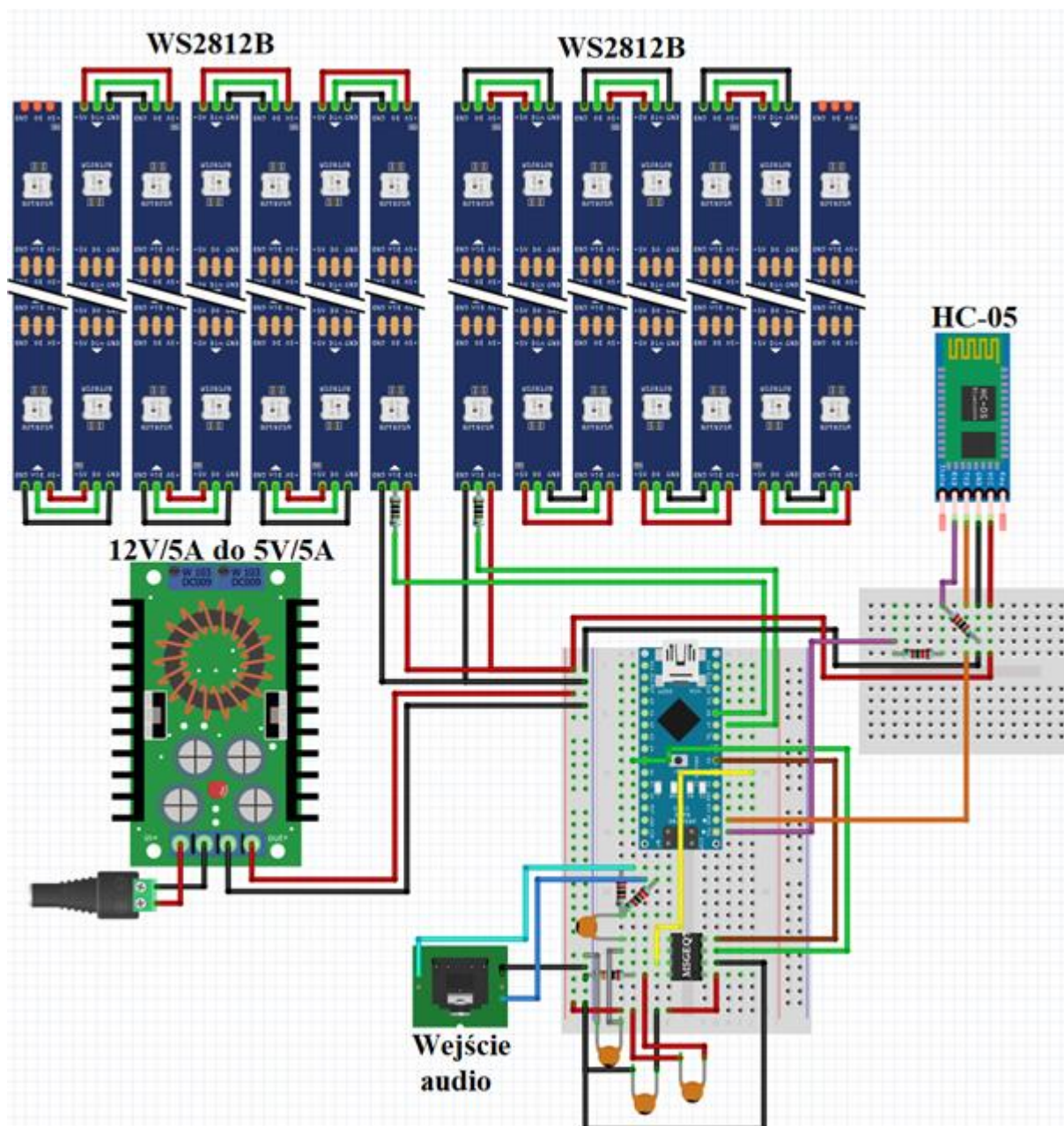
Rys. 23 Wyświetlacz widma dźwięku – schemat blokowy.

4.2.2. Schemat połączeń elementów

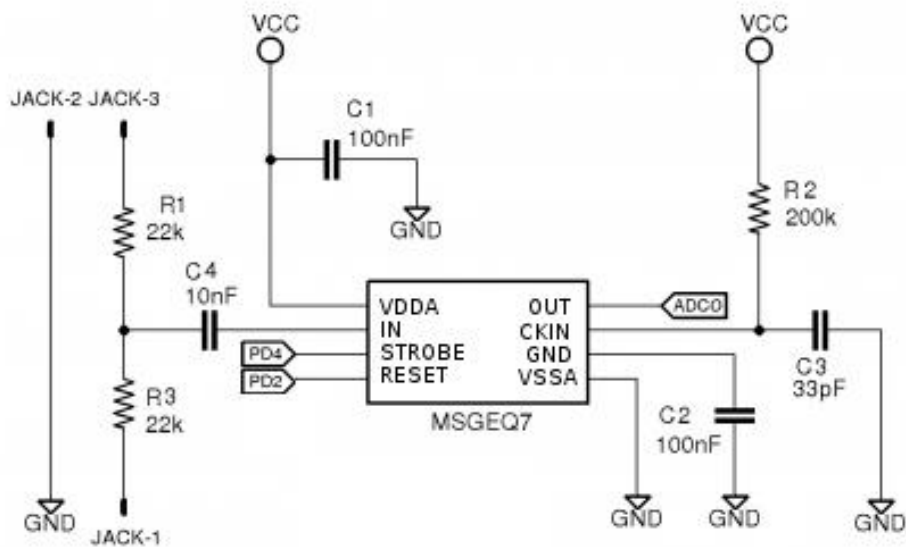
W celu uproszczenia schematu wyświetlacza widma dźwięku przedstawiono połączenie modułu Bluetooth HC-05 na dodatkowej płytce stykowej. Realizacja fizyczna nie zawiera jej, ponieważ całą elektronikę udało zmieścić się bez konieczności stosowania kolejnego elementu.

Na schemacie połączeń całego układu warto zauważyć, jeden układ MSGEQ7 oraz 14 kolumn diod LED WS2812B. Dzięki zastosowaniu takiego rozwiązania uzyskano lustrzane odbicie widma dźwięku. Dodatkowo, tak jak w przypadku inteligentnego budynku, pierwsze diody w szeregu zabezpieczono przed przepaleniem wykorzystując w tym celu rezystory $1k\Omega$.

Dokładne komponenty oraz sposób podłączenia układu scalonego MSGEQ7 przedstawiono na rysunku 25.



Rys. 24 Wyświetlacz widma dźwięku – schemat połączeń.



Rys. 25 Wyświetlacz widma dźwięku – podłączenie MSGEQ7 [37].

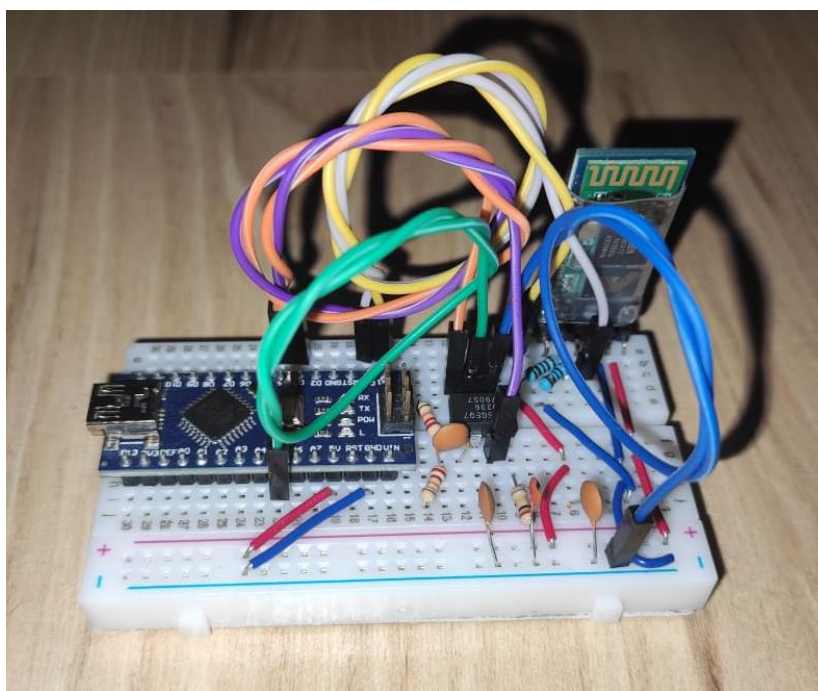
4.2.3. Montaż elementów

Przygotowując elementy do montażu w pierwszej kolejności sprawdzono poprawność konstrukcji urządzenia. Po weryfikacji wymiarów, odmierzone i podzielono taśmę diod LED WS2812B po 13 w kolumnie. Dostosowano, również długość maskownic. Następnie wykonano miejsca na poprowadzenie przewodów. Zlutowany układ przedstawiono na rysunku 26.



Rys. 26 Wyświetlacz widma dźwięku – lutowanie diod WS2812B.

Kolejnym etapem była realizacja obwodu elektrycznego. Wykonano go zgodnie z wcześniej przedstawionym schematem montażowym. Jedynym odstępstwem jest brak dodatkowej płytki stykowej, ponieważ moduł Bluetooth HC-05 udało się zmieścić wraz z innymi elementami elektronicznymi. W wyniku tego działania ograniczono ilość zajmowanego miejsca.



Rys. 27 Wyświetlacz widma dźwięku – zrealizowany obwód elektryczny.

Gotowy projekt podłączono do wyjścia liniowego wzmacniacza audio. Sposób ten pozwolił uniezależnić się od głośności odsłuchu utworów muzycznych. Tradycyjny sposób rozgałęzienia na przewodach pomiędzy wzmacniaczem a głośnikami wpłynął by negatywnie ilość zapalanych diod.



Rys. 28 Wyświetlacz widma dźwięku – gotowe urządzenie.

5. Oprogramowanie

Do realizacji oprogramowania mikrokontrolerów wykorzystano środowisko programistyczne Arduino IDE. Jest to aplikacja wieloplatformowa (działająca na systemach operacyjnych: Windows, macOS oraz Linux). Używana jest do tworzenia oprogramowania oraz wgrywania programów do fizycznych modułów. Wykorzystuje języki programowania C oraz C++.

Aplikacja mobilna została zaprojektowana i wykonana za pomocą oprogramowania MIT App Inventor – zintegrowane środowisko programistyczne umożliwiające tworzenie oprogramowania na urządzenia mobilne z systemem operacyjnym Android. Dodatkowo producent zapewnił możliwość bezpośredniego pobrania, stworzonego oprogramowania, za pomocą kodu QR przy użyciu bliźniaczej aplikacji MIT AI2 Companion.

5.1. Inteligentny budynek

System inteligentnego domu wymaga obsługi wielu urządzeń. Ze względu na integrację wszystkich modułów, aplikacja mobilna musi posiadać możliwość obsługi każdego z nich.

5.1.1. Konfiguracja

Na początku konfiguracji dołączono odpowiednie biblioteki, w celu dodania do programu odpowiednich metod, umożliwiających skonstruowanie działającego systemu. Wykorzystanie wbudowanych funkcji pozwoliło na poprawę czytelności kodu oraz przyspieszyło proces tworzenia oprogramowania.

Następnie zdefiniowano poszczególne piny wejścia i wyjścia oraz określono ilość diod LED WS2812B, wykorzystywanych w oświetleniu budynku. Zarówno na parterze makiety budynku, jak i na piętrze liczba pikseli jest ekwiwalentna.

Zainicjalizowano, także obiekty odpowiadające za sterowanie światłem oraz bramą garażową.

```
//Fan
#define FAN 6

//Light
#define LED_COUNT 21
#define LIGHT_UP 7
#define LIGHT_DOWN 12

//Alarm
#define ALARM 8
#define RED_ALARM 9
#define GREEN_ALARM 10

//Garage doors
#define GARAGE_DOORS 3

#include <Adafruit_NeoPixel.h>
#include <Servo.h>

Adafruit_NeoPixel lightUp = Adafruit_NeoPixel(LED_COUNT, LIGHT_UP, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel lightDown = Adafruit_NeoPixel(LED_COUNT, LIGHT_DOWN, NEO_GRB + NEO_KHZ800);

Servo garageDoors;
```

Rys. 29 Inteligentny budynek – konfiguracja programu.

Procedura „*setup*” rozpoczyna program. Wykonuje się jedynie raz w momencie uruchomienia modułu. Jej przeznaczeniem jest konfiguracja programu. W tej części oprogramowania

ustawiono odpowiednią prędkość transmisji danych, wynoszącą 9600bps oraz tryb pracy pinów: wejścia lub wyjścia.

```
void setup() {  
    Serial.begin(9600);  
  
    //Light  
    lightUp.begin();  
    lightUp.show();  
    lightDown.begin();  
    lightDown.show();  
  
    //Fan  
    pinMode(FAN, OUTPUT);  
  
    //Alarm  
    pinMode(ALARM, INPUT);  
    pinMode(RED_ALARM, OUTPUT);  
    pinMode(GREEN_ALARM, OUTPUT);  
  
    //Garage doors  
    garageDoors.attach(GARAGE_DOORS);  
    garageDoors.write(0);  
    delay(150);  
    garageDoors.writeMicroseconds(1500);  
    garageDoors.detach();  
}
```

Rys. 30 Inteligentny budynek – funkcja startowa.

5.1.2. Główny algorytm

Procedura „loop” jest to nieskończona pętla, w której należy umieścić logikę programu. Polecenia w niej zawarte będą wykonywać się dopóki Arduino będzie zasilane.

Napisane oprogramowanie wykorzystuje instrukcje wielokrotnego wyboru – „switch”. Warunkiem wykonania się odpowiedniego fragmentu żądania jest wartość dostarczana przez funkcję „userInput”. Zwraca ona liczbę od 0 do 11, odebraną z aplikacji mobilnej, w wyniku działania modułu Bluetooth HC-05. Każda z nich, odpowiada za wykonanie się innej instrukcji. Przykładowo odebranie cyfry „2”, wywoła funkcję „turnOnFan”, która włączy wentylację. Wartości niedozwolone, czy też wykraczające poza przedział nie wpłyną na działanie systemu.

Monitoring bezpieczeństwa inteligentnego budynku, po włączeniu wymaga ciągłego kontrolowania wykrycia obecności w danym pomieszczeniu. W tym celu funkcja główna zawiera

instrukcje sprawdzające, czy alarm jest włączony. Jeśli oprogramowanie potwierdza stan gotowości czujnika, następuje nieustanna praca urządzenia sygnalizowana włączeniem czerwonej diody LED. W momencie wykrycia ruchu zostają zmienione stany logiczne dwóch pinów. Skutkują one włączeniem diody zielonej oraz wyłączeniem diody czerwonej.

```
void loop() {
  switch (userInput()) {
    case 0:
      turnOnLights();
      break;
    case 1:
      turnOffLights();
      break;
    case 2:
      turnOnFan();
      break;
    case 3:
      turnOffFan();
      break;
    case 4:
      turnOnAlarm();
      break;
    case 5:
      turnOffAlarm();
      break;
    case 6:
      openGarage();
      break;
    case 7:
      closeGarage();
      break;
    case 8:
      increaseLightingPower();
      break;
    case 9:
      decreaseLightingPower();
      break;
    case 10:
      increaseFanSpeed();
      break;
    case 11:
      decreaseFanSpeed();
      break;
    default:
      break;
  }

  //Alarm - continuous scanning when is ON
  if(getIsAlarmOn()) {
    scanning();
  }
}
```

Rys. 31 Inteligentny budynek – główny algorytm programu.

5.1.3. Kluczowe funkcje

W celu oddzielenia logiki programu od jego kluczowych funkcji, zdecydowano umieścić je w osobnych plikach źródłowych z rozszerzeniem „.ino”. Jest to format plików oprogramowania stworzonego za pomocą Arduino IDE.

Na skutek tego podziału poprawiono czytelność kodu oraz zwiększono możliwości na dalszą rozbudowę w przyszłości. Poniżej przedstawiono fragmenty zawartości stworzonych plików wraz z opisem ich najważniejszych funkcji.

- Alarm.ino – funkcja „*scanning*” zawiera logikę programu pozwalającą na detekcję ruchu. Zależnie od wartości „*true*” lub „*false*”, zwracanej przez funkcję „*isDetectMove*” ustawione zostaną odpowiednie stany napięcia na wyjściach pinów diod LED.

```
void scanning() {
    if(isDetectMove()) {
        turnOnRedAlarm();
    } else {
        turnOnGreenAlarm();
    }
}

bool isDetectMove() {
    return digitalRead(ALARM);
}

void turnOnRedAlarm() {
    digitalWrite(REDA_ALARM, HIGH);
    digitalWrite(GREEN_ALARM, LOW);
}

void turnOnGreenAlarm() {
    digitalWrite(GREEN_ALARM, HIGH);
    digitalWrite(REDA_ALARM, LOW);
}
```

Rys. 32 Inteligentny budynek – oprogramowanie alarmu.

- GarageDoors.ino – logika oprogramowania tej funkcji polega na ustawianiu odpowiedniej wartości na linii danych serwomechanizmu, w celu obrotu orczyka do którego przymocowana została brama garażowa.

```

void openGarage() {
    garageDoors.attach(GARAGE_DOORS);

    garageDoors.write(180);
    delay(300);
    garageDoors.writeMicroseconds(1500);

    garageDoors.detach();
}

```

Rys. 33 Inteligentny budynek – oprogramowanie bramy garażu.

- Fan.ino – „isFanTurnOn” wykorzystywany jest do sprawdzenia, czy wentylator jest włączony podczas próby zwiększenia lub zmniejszenia jego prędkości obrotowej. Jeżeli układ jest włączony i nie przekracza maksymalnej wartości prędkości obrotowej, następuje jej zmiana o domyślną wartość 5 oraz ustawienie nowego stanu napięcia na pinie wentylatora.

```

void turnOnFan() {
    isFanTurnOn = true;
    analogWrite(FAN, fanSpeed);
}

void increaseFanSpeed() {
    if(fanSpeed < MAX_FAN_SPEED && isFanTurnOn)
        fanSpeed += 5;
    analogWrite(FAN, fanSpeed);
}

```

Rys. 34 Inteligentny budynek – fragment oprogramowania wentylatora.

- Lights.ino – niżej wymieniona funkcja odpowiada za włączenie oświetlenia w inteligentnym budynku. Na początku zostaje ustawiona wartość „true” dla globalnej zmiennej „isLightTurnOn”. Wykorzystywana jest ona do weryfikacji, czy oświetlenie jest włączone w momencie próby zwiększenia lub zmniejszenia natężenia światła.

```

void turnOnLights() {
    isLightTurnOn = true;

    for(int i = 0; i < LED_COUNT; ++i) {
        lightUp.setPixelColor(i, lightUp.Color(lightningPower, lightningPower, lightningPower));
        lightDown.setPixelColor(i, lightDown.Color(lightningPower, lightningPower, lightningPower));
    }

    lightUp.show();
    lightDown.show();
}

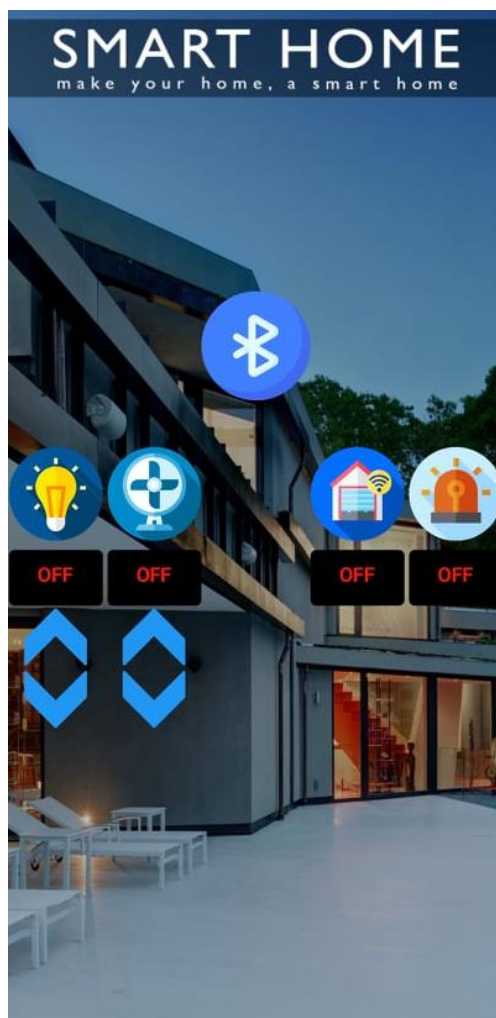
```

Rys. 35 Inteligentny budynek – fragment oprogramowania oświetlenia.

5.1.4. Aplikacja mobilna

Proces tworzenia aplikacji mobilnej na telefon z systemem Andriod rozpoczęto od wyboru tła. Po analizie dostępnych grafik, zdecydowano się na obraz zamieszczony na stronie: fone-walls.com [42]. Elementy graficzne interfejsu użytkownika wybrano, spośród gotowych rozwiązań dostępnych na stronie: www.flaticon.com [43].

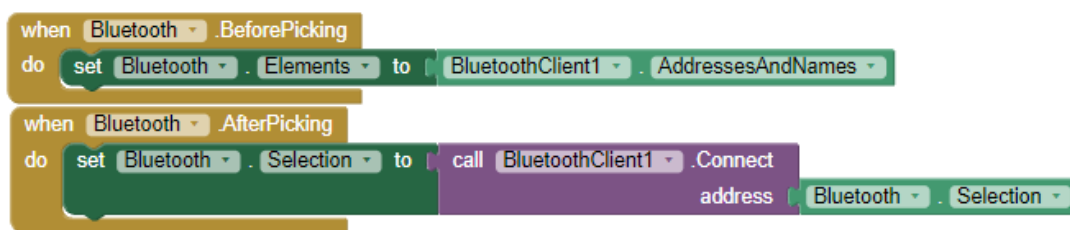
Gotowa aplikacja prezentuje się w następujący sposób:



Rys. 36 Inteligentny budynek – interfejs użytkownika.

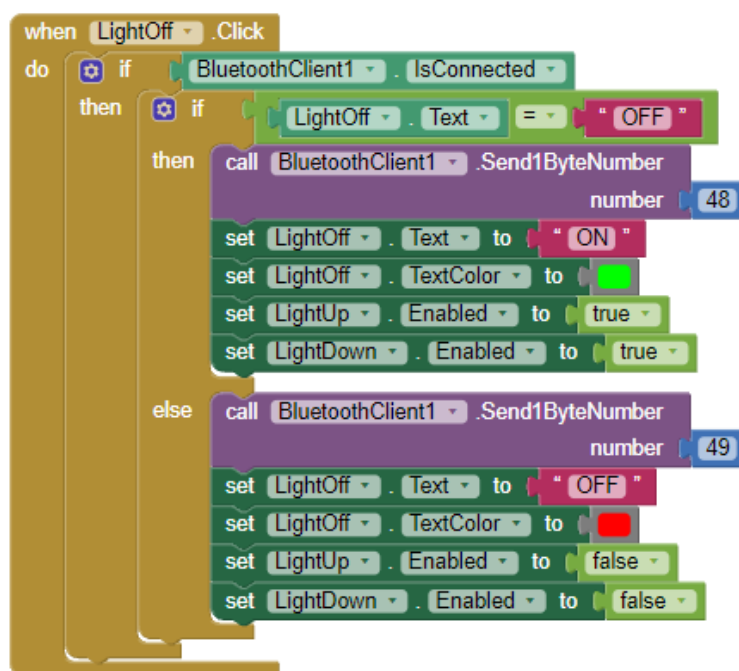
Na środku ekranu znajdują się ikona Bluetooth służąca do wyboru i ustawienia połączenia z wybranym urządzeniem.

Każdemu modułowi w systemie odpowiada stosowna ikona. Tryb urządzeń sygnalizowany jest za pomocą informacji wyświetlanych na przyciskach. Stan „OFF”, oznacza że podzespół jest wyłączony, z kolei „ON” – włączony. Poszczególne tryby można zmieniać na stan przeciwny za pomocą wciśnięcia odpowiadającym im przycisków. Dodatkowo system został wyposażony w możliwość regulacji natężenia oświetlenia oraz prędkości obrotowej wentylatora. Służą do tego strzałki umieszczone pod stosownymi włącznikami.



Rys. 37 Inteligentny budynek – komunikacja Bluetooth.

Komunikacje Bluetooth zapewnia fragment oprogramowania przedstawiony na rysunku 37. Pierwszy element, odpowiedzialny jest za wyświetlenie listy dostępnych i kompatybilnych urządzeń po wciśnięciu ikony Bluetooth. Drugi z kolei ustala połączenie pomiędzy aplikacją a wybranym urządzeniem.



Rys. 38 Inteligentny budynek – fragment logiki komunikacji.

Fragment logiki oprogramowania przedstawia obsługę oświetlenia. Na początku sprawdzany jest stan połączenia Bluetooth. Jeżeli komunikacja jest zapewniona następuje zmiana trybu urządzenia na przeciwny. Wysyłana jest wartość odpowiadająca za wybrane żądanie do Arduino Uno oraz wyświetlana jest informacja zwrotna, czy praca urządzenia została aktywowana lub dezaktywowana. W celu lepszej wizualizacji stanu, tekst posiada adekwatny kolor. Ponadto zabezpieczono program przed regulacją mocy oświetlenia, w momencie trybu wyłączenia.

5.2. Wyświetlacz widma dźwięku

Głównym zadaniem oprogramowania wyświetlacza widma dźwięku jest obsługa układu scalonego MSGEQ7 oraz adresowalnych diod LED WS2812B za pomocą Arduino Nano. Należało, również zapewnić komunikację Bluetooth, aby uzyskać możliwość zmiany barwy światła.

5.2.1. Konfiguracja

Tworzenie oprogramowania rozpoczęto od dołączenia do programu odpowiednich bibliotek. W tym celu, aby zarządzać diodami LED WS2812B wykorzystano bibliotekę „Adafruit NeoPixel”, która została specjalnie przystosowana do kontrolowania pikseli za pomocą jednej linii danych. Dodatkowo komunikacja Bluetooth pomiędzy modulem, a wyświetlaczem widma dźwięku wymagała użycia biblioteki „SoftwareSerial”. Pozwoliła ona na otrzymywanie żądań z aplikacji mobilnej.

W kolejnym etapie stworzono siedmioelementową tablicę, aby zapisywać w niej wartości z wejścia analogowego. Zdefiniowano, również odpowiednie piny oraz zainicjalizowano wartości domyślne poszczególnych kolorów.

Omówione działania przedstawiono na rysunku poniżej.

```

#include <Adafruit_NeoPixel.h>
#include <SoftwareSerial.h>

#define STORBE_PIN 4
#define RESET_PIN 3
#define OUT_PIN A5
#define WS2812_PIN_R 7
#define WS2812_PIN_L 8

int level[7];

#define ASCII_OFFSET 48

#define LED_MAX_VALUE 240
#define LED_MIN_VALUE 5
#define LED_VALUE_STEP 5

int8_t color_red = 10;
int8_t color_green = 10;
int8_t color_blue = 10;

SoftwareSerial BTserial(0,1);

Adafruit_NeoPixel stripR = Adafruit_NeoPixel(91, WS2812_PIN_R, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel stripL = Adafruit_NeoPixel(91, WS2812_PIN_L, NEO_GRB + NEO_KHZ800);

```

Rys. 39 Wyświetlacz widma dźwięku – konfiguracja programu.

W funkcji startowej programu ponownie, jak w inteligentnym budynku, ustawiono prędkość transmisji danych, wynoszącą 9600bps. Następnie wykonano wywołanie metod konfiguracyjnych na obiektach klasy „Adafruit_NeoPixel” oraz zainicjalizowano funkcje i stany logiczne pinów. W kolejnym etapie zresetowano MSGEQ7, zachowując odstępy czasowe z noty katalogowej układu.

```

void setup()
{
    Serial.begin  (9600);
    BTserial.begin(9600);

    stripR.begin();
    stripR.show();

    stripL.begin();
    stripL.show();

    pinMode      (STORBE_PIN,  OUTPUT);
    pinMode      (RESET_PIN,   OUTPUT);
    pinMode      (OUT_PIN,     INPUT);

    digitalWrite (RESET_PIN,   LOW);
    digitalWrite (STORBE_PIN,  LOW);
    delay        (1);

    digitalWrite (RESET_PIN,   HIGH);
    delay        (1);
    digitalWrite (RESET_PIN,   LOW);
    digitalWrite (STORBE_PIN,  HIGH);
    delay        (1);
}

```

Rys. 40 Wyświetlacz widma dźwięku – funkcja startowa.

5.2.2. Główny algorytm

Główna logika programu została umieszczona w funkcji „*loop*”. W pierwszym etapie algorytm sprawdza, czy użytkownik zmienił kolor w aplikacji mobilnej i zależnie od rezultatu podejmuje odpowiednie działanie. Część tego oprogramowania szczegółowo opisano w dalszej części pracy.

W kolejnym fragmencie kodu wybierane są kolejne wejścia multipleksera MSGEQ7, wskutek zmian stanów pinu „Storbe”. Pobrane dane z wejścia analogowego zachowane zostają w tablicy siedmioelementowej. Warto zaznaczyć, że pomiędzy kontrolą pinu „Storbe”, zachowano odstęp czasowy z pewnym zapasem, w celu zapewnienia bezpieczeństwa oprogramowania.

Wywołania funkcji „*clear*”, wymazują konfiguracje znajdujące się w obiektach.

Kolejna pętla „*for*”, zawierająca funkcję „*setBar*”, definiuje i zapisuje w obiektach ilość zapalonych pikseli.

Ostatni etap polega na wywołaniu funkcji „show” w celu włączenia zdefiniowanych diod WS2812B.

Oczekiwanie 10000 mikrosekund na ponowne wykonanie się nieskończonej pętli programu, zastosowano w celu ograniczenia prędkości migania oświetlenia.

```
void loop()
{
    changeColor();

    for (int i = 0; i < 7; i++) {
        digitalWrite (STORBE_PIN, LOW);
        delayMicroseconds (100);

        level[i] =      analogRead (OUT_PIN);

        digitalWrite (STORBE_PIN, HIGH);
        delayMicroseconds (100);
    }

    stripR.clear();
    stripL.clear();

    for (int i = 0; i < 7; i++)
    {
        setBar(i, level[i]);
    }

    stripR.show();
    stripL.show();

    delayMicroseconds(10000);
}
```

Rys. 41 Wyświetlacz widma dźwięku – główny algorytm.

5.2.3. Kluczowe funkcje

Jedną z kluczowych ról w oprogramowaniu pełni funkcja „setBar” (rys 42.). Odpowiada za określenie oraz zapisanie, które piksele mają zostać włączone. Wywołując funkcje w pętli głównej programu przekazujemy jej argumenty „i” oraz „level”, oznaczających kolejno numer kolumny diod WS2812B oraz odczyt amplitudy dźwięku wysyłanego z układu MSGEQ7 do analogowego wejścia Arduino.

```

void setBar(byte i, int level)
{
    byte level_mapped = map(level, 100, 950, 0, 13);

    for (int y = 0; y < level_mapped; y++)
    {
        stripR.setPixelColor((i*13)+y, stripR.Color(color_red, color_green, color_blue));
        stripL.setPixelColor((i*13)+y, stripL.Color(color_red, color_green, color_blue));
    }
}

```

Rys. 42 Wyświetlacz widma dźwięku – ustawienie koloru poszczególnych pikseli.

Warto zaznaczyć, że otrzymane wartości „*level*” należą do przedziału $\langle 0, 1024 \rangle$, dlatego pierwszym etapem jest przeskalowanie argumentu do wartości z przedziału $\langle 0, 13 \rangle$ odpowiadającemu za:

- 0 – wyłączenie wszystkich pikseli w danej kolumnie,
- od 1 do 13 – włączenie konkretnej liczby pikseli w danej kolumnie.

W tym celu posłużono się wbudowaną funkcją „*map*”. Celowo ograniczono przedział skalowania do zakresu pomiędzy 100, a 950, aby zminimalizować szумы oraz błędy wynikające z niedokładności układu scalonego MSGEQ7.

Kolejnym etapem jest wykonanie pętli „*for*” w ilości zdefiniowanej przez liczbę zapalanych diod WS2812B. Wywołanie funkcji „*setPixelColor*” na obiektach „*stripR*” oraz „*stripL*”, pozwala na zapis podanych wartości kolorów w pamięci programu.

Posiadając informacje, że w każdej kolumnie umieszczone jest 13 pikseli, możemy przemnożyć tę wartość przez numer kolumny. W ten sposób uzyskujemy pierwszy, potencjalny piksel do ustawienia. Pojedynczy wybór piksela polega na dodaniu do wcześniej obliczonej liczby zmiennej „*y*”, której wartość wynosi numerowi iteracji w pętli. W wyniku otrzymujemy informacje, którą diodę WS2812B należy ustawić.

Podanie do funkcji „*Color*”, argumentów zawierających dane na temat jasności odpowiednich diod RGB, określi kolor odpowiedniego piksela. W wyniku tych działań wywołanie metody „*show*”, umieszczonej w głównej części programu skonfigurowane diody zostaną włączone.

Zmiana koloru odbywa się za pomocą danych dostarczanych do modułu z aplikacji mobilnej. Jeżeli program wykryje, dostępne dane na pinie RX, następuje ich zapis do zmiennej „*input*”. W konsekwencji na jej podstawie rozpoczyna się analiza, którą wartość koloru należy zmniejszyć, bądź zwiększyć oraz czy podane żądanie nie przekroczy ustawionego minimalnego

i maksymalnego zakresu jasności. Funkcja obsługująca zmianę koloru przedstawiona została na rysunku 43.

```
void changeColor()
{
    if(Serial.available() > 0) {
        int input = Serial.read();

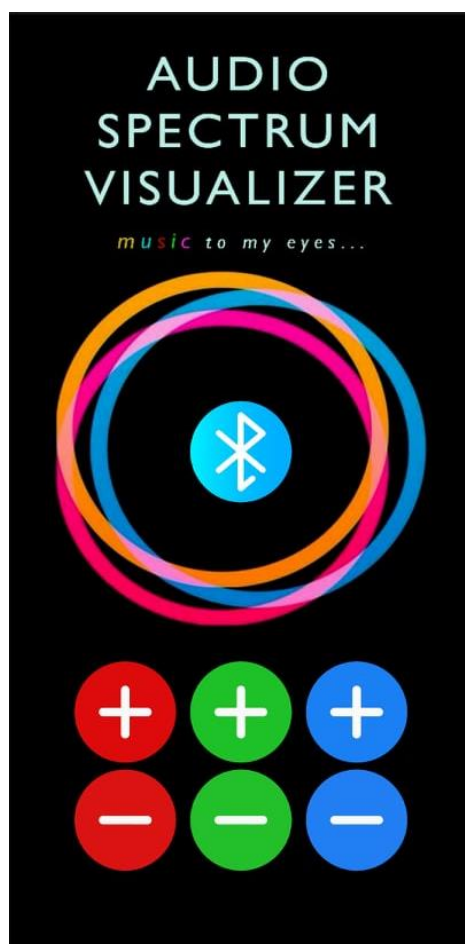
        if(color_green <= LED_MAX_VALUE && input == 1)
        {
            color_green += LED_VALUE_STEP;
        }
        else if(color_red <= LED_MAX_VALUE && input == 2)
        {
            color_red += LED_VALUE_STEP;
        }
        else if(color_blue <= LED_MAX_VALUE && input == 3)
        {
            color_blue += LED_VALUE_STEP;
        }
        else if(color_green >= LED_MIN_VALUE && input == 4)
        {
            color_green -= LED_VALUE_STEP;
        }
        else if(color_red >= LED_MIN_VALUE && input == 5)
        {
            color_red -= LED_VALUE_STEP;
        }
        else if(color_blue >= LED_MIN_VALUE && input == 6)
        {
            color_blue -= LED_VALUE_STEP;
        }
    }
}
```

Rys. 43 Wyświetlacz widma dźwięku – ustawienie wartości koloru diod RGB, na podstawie danych z aplikacji mobilnej.

5.2.4. Aplikacja mobilna

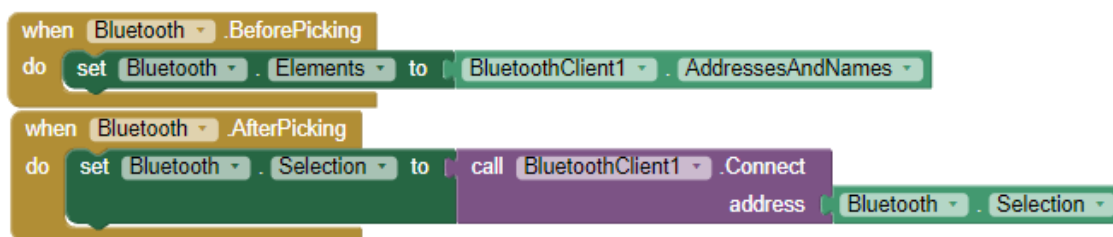
Pierwszym podjętym krokiem, był dobór elementów graficznych interfejsu użytkownika. Tło aplikacji stworzono w aplikacji GIMP – bezpłatne oprogramowanie do edycji graficznej, natomiast w celu wyboru grafiki przycisków, skorzystano z gotowych, darmowych rozwiązań

zamieszczonych na stronie: www.flaticon.com [43]. Jedyna modyfikacja polegała na zmianie kolorów ikon, zawierających symbole plus oraz minus. Wymienione elementy rozmieszczono w głównej części aplikacji, tworząc interfejs użytkownika.



Rys. 44 Wyświetlacz widma dźwięku – interfejs użytkownika.

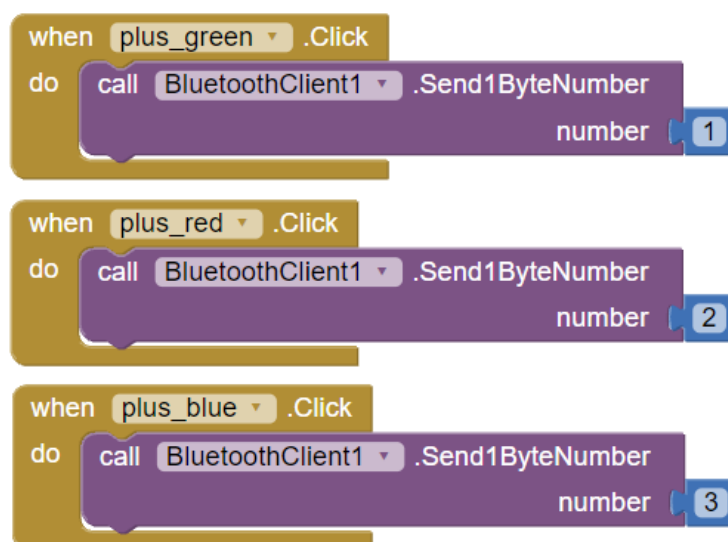
Po rozmieszczeniu i dostosowaniu rozmiaru przycisków, dodano ich funkcjonalności. Ikonę z symbolem Bluetooth oprogramowano, aby po wciśnięciu umożliwiła wybór i połączenie z modułem Bluetooth HC-05. W tym celu dodano do elementu logikę przedstawioną na rysunku 45.



Rys. 45 Wyświetlacz widma dźwięku – komunikacja Bluetooth.

Fragment głównej funkcji aplikacji mobilnej, polegającej na zmniejszaniu lub zwiększaniu natężenia światła przez poszczególne diody RGB prezentuje rysunek 46. Koncept oparty jest na wysyłaniu do modułu Arduino Nano odpowiednich wartości oznaczających kolejno:

- „1” – zwiększ natężenie diod zielonych,
- „2” – zwiększ natężenie diod czerwonych,
- „3” – zwiększ natężenie diod niebieskich,
- „4” – zmniejsz natężenie diod zielonych,
- „5” – zmniejsz natężenie diod czerwonych,
- „6” – zmniejsz natężenie diod niebieskich.



Rys. 46 Wyświetlacz widma dźwięku – fragment głównej logiki aplikacji.

6. Testy

W trakcie wykonywania testów wymagany był telefon z systemem operacyjnym Android, ponieważ to na ten system została opracowana aplikacja mobilna. Dodatkowo smartfon musiał posiadać moduł Bluetooth w celu przeprowadzenia łączności, pomiędzy nim, a urządzeniami. Podane wymagania spełnił prywatny smartfon Xiaomi Mi Note 10 Lite. Model ten posiada system operacyjny Android 11 oraz moduł Bluetooth w wersji 5.0.

W dalszej części pracy przetestowano zaprojektowane urządzenia, a wyniki tych działań przedstawiono w tabelach wraz z rezultatami.

Ze względu na zastosowanie dwóch modułów Bluetooth HC-05, pochodzących od różnych sprzedawców, dokonano testów łączności zarówno dla inteligentnego budynku oraz wyświetlacza widma dźwięku. Ponadto z przestrzeni testów usunięto elementy, które mogłyby wpłynąć negatywnie na jakość komunikacji, takie jak router sieci WiFi, czy też przeszkody znajdujące się na wyznaczonej drodze.

6.1. Inteligentny budynek

Testowanie urządzenia rozpoczęto od sprawdzenia funkcjonalności systemu, znajdujących się w inteligentnym budynku. W tym celu zainstalowano stworzone oprogramowanie na smartfonie i wysyłano za jego pomocą żądania. Znak „–” w komórce tabeli oznacza, że dany test nie dotyczy konkretnego podzespołu.

Tab. 1 Inteligentny budynek – test funkcjonalności urządzenia.

Podzespół systemu	Włączenie funkcjonalności	Wyłączenie funkcjonalności	Zwiększenie natężenia	Zmniejszenie natężenia
Alarm	Pozytywny	Pozytywny	–	–
Brama garażowa	Pozytywny	Pozytywny	–	–
Wentylacja	Pozytywny	Pozytywny	Pozytywny	Pozytywny
Oświetlenie	Pozytywny	Pozytywny	Pozytywny	Pozytywny

Po sprawdzeniu funkcjonalności wykonano testy komunikacji modułu Bluetooth HC-05 z aplikacją mobilną. W tym celu ponownie połączono się z urządzeniem, a następnie spraw-

dzano jego funkcjonalność stopniowo się oddalając. Wskutek tego sprawdzono stabilność połączenia zależnie od odległości modułu i smartfona. Test został wykonany w warunkach domowych (z ograniczoną ilością miejsca), a jego rezultaty przedstawiono w tabeli poniżej.

Tab. 2 Inteligentny budynek – test komunikacji Bluetooth z urządzeniem.

Odległość [m]	Poprawna funkcjonalność
1	Pozytywny
2	Pozytywny
3	Pozytywny
4	Pozytywny
5	Pozytywny
6	Pozytywny
7	Pozytywny
8	Pozytywny
9	Pozytywny
10	Pozytywny
11	Pozytywny
12	Pozytywny
13	Negatywny

6.2. Wyświetlacz widma dźwięku

Przetestowanie wyświetlacza widma dźwięku polegało na wykonaniu trzech testów:

- analizie poprawności wyświetlanych diod LED WS2812B w poszczególnych kolumnach,
- sprawdzeniu działania aplikacji mobilnej,
- ocenie komunikacji pomiędzy modulem Bluetooth HC-05 oraz urządzeniem.

Proces rozpoczęto od pobrania i zainstalowania aplikacji mobilnych: dedykowanej dla wyświetlacza widma dźwięku oraz drugiej, pozwalającego na generowanie wybranych częstotliwości – Pro Audio Tone Generator [44].

Połączono smartfona poprzez komunikację Bluetooth z urządzeniem, a następnie wygenerowano sygnał sinusoidalny. Zmieniając płynnie częstotliwość sygnału oraz sprawdzając ilość zapalanych diod LED WS2812B, oceniono poprawność działania urządzenia. Wyniki dla lewego oraz prawego zakresu częstotliwości zapisano w tabeli poniżej.

Tab. 3 Wyświetlacz widma dźwięku – test działania diod LED WS2812B w kolumnach.

Częstotliwość [Hz]	Wyświetlacz widma dźwięku	
	Lewy zakres	Prawy zakres
63	Pozytywny	Pozytywny
160	Pozytywny	Pozytywny

400	Pozytywny	Pozytywny
1000	Pozytywny	Pozytywny
2500	Pozytywny	Pozytywny
6250	Pozytywny	Pozytywny
16000	Pozytywny	Pozytywny

Test aplikacji mobilnej przeprowadzono w trakcie odsłuchów utworów muzycznych. Wybrano odpowiednie kompozycje, a następnie zmniejszono wszystkie kolory, aż do momentu wyłączenia diod. Po tym zabiegu stopniowo wysyłano pojedyncze żądania w odstępach czasowych, analizując wizualne rezultaty tego działania. Na początku skupiono się na kolorze czerwonym, aż do osiągnięcia maksymalnego natężenia. Po osiągnięciu tej wartości rozpoczęto obniżanie mocy, aż do zera. Po wykonaniu operacji i zapisaniu wyniku, ponowiono próbę dla pozostałych kolorów. Rezultat zamieszczono w tabeli 4.

Tab. 4 Wyświetlacz widma dźwięku – test aplikacji mobilnej.

Żądanie aplikacji mobilnej	Kolor diod LED WS2812B		
	Czerwony	Zielony	Niebieski
Zwiększenie natężenia	Pozytywny	Pozytywny	Pozytywny
Zmniejszenie natężenia	Pozytywny	Pozytywny	Pozytywny

Ostatni etap polegał na teście komunikacji Bluetooth urządzenia ze smartfonem. Test przeprowadzono w warunkach domowych. W trakcie próby stopniowo zwiększano odległość o 1 metr w linii prostej, a następnie sprawdzano reakcje na wysyłane komendy zmiany kolorów. Wyniki zanotowano w tabeli poniżej.

Tab. 5 Wyświetlacz widma dźwięku – test komunikacji Bluetooth z urządzeniem.

Odległość [m]	Poprawna funkcjonalność
1	Pozytywny
2	Pozytywny
3	Pozytywny
4	Pozytywny
5	Pozytywny
6	Pozytywny
7	Pozytywny
8	Pozytywny
9	Pozytywny
10	Pozytywny
11	Pozytywny
12	Negatywny

7. Podsumowanie i wnioski

W niniejszej pracy inżynierskiej został przedstawiony przebieg budowy modelu inteligentnego budynku wraz z wyświetlaczem widma dźwięku wchodzącego w jego skład.

W pierwszej części zawarto szczegóły na temat wymagań pracy. Sprecyzowano cele, które stopniowo realizowano wraz z postępem projektu.

Rozpoczęto od badania rynku inteligentnych rozwiązań oraz chęci ich posiadania przez klientów. W rezultacie uświadomiono rosnącą popularność systemów automatyki i monitoringu budynków. Ponadto zaprezentowano szacowane koszty modernizacji, które w dużej części zależą od zaawansowania systemów.

Kolejne rozdziały opisują proces realizacji fizycznych urządzeń oraz aplikacji mobilnych. Zaprezentowano dobrane podzespoły systemu oraz stworzono schematy elektroniczne. Na ich podstawie przeprowadzono montaż urządzeń, które zaprogramowano zgodnie z wytycznymi.

Cały proces poddano dokładnej analizie. Projekt został przetestowany pod względem funkcjonalności oraz jakości połączenia pomiędzy smartfonem a urządzeniami. Otrzymane wyniki potwierdzają poprawność działania urządzeń. Z rezultatów testów, również wynika, że maksymalna odległość komunikacji z modułem Bluetooth HC-05 jest większa, niż informuje producent. Zależy ona zapewne od różnych warunków otoczenia i użytego sprzętu.

Wszystkie założone cele udało się pomyślnie zrealizować. W trakcie realizacji pracy w sposób praktyczny utrwalono zdobytą wiedzę. Umiejętności z zakresu elektroniki i programowania, którą uzyskano w trakcie studiów, okazała się niezbędna do ukończenia projektu.

Spis ilustracji

Rys. 1 Komponenty budynku wymagające automatyzacji [2].	7
Rys. 2 Inteligentne gniazdko WiFi Gosung SP1 [6].	8
Rys. 3 Zestaw sześciu inteligentnych urządzeń marki Fibaro [7].	9
Rys. 4 Ilustracja poglądowa modułu Arduino Uno [16].	11
Rys. 5 Ilustracja poglądowa modułu Arduino Nano [18].	11
Rys. 6 Ilustracja poglądowa mikro serwomechanizmu SG90 [19].	12
Rys. 7 Ilustracja poglądowa czujnika PIR HC-SR501 [22].	13
Rys. 8 Ilustracja poglądowa modułu Bluetooth HC-05 [27].	14
Rys. 9 Ilustracja poglądowa taśmy LED WS2812B [32].	15
Rys. 10 Przykładowe dostarczanie danych do diod LED WS2812B [33].	16
Rys. 11 Ilustracja poglądowa układu scalonego MSGEQ7 [34].	16
Rys. 12 Przykład działania układu MSGEQ7 [34].	17
Rys. 13 Przykładowe rozwiązanie detekcji szczytu [36].	17
Rys. 14 Schemat układu MSGEQ7.	18
Rys. 15 Diagram czasowy MSGEQ7.	19
Rys. 16 Ilustracja poglądowa wentylatora DFR0332 [39].	20
Rys. 17 Ilustracja poglądowa przetwornicy impulsowej DC-DC step-down [40].	20
Rys. 18 Inteligentny budynek – schemat blokowy systemu.	22
Rys. 19 Inteligentny budynek – schemat połączeń systemu.	23
Rys. 20 Schemat podłączenia Bluetooth HC-05 z Arduino Uno [41].	23
Rys. 21 Makieta inteligentnego budynku.	25
Rys. 22 Gotowy inteligentny budynek.	26
Rys. 23 Wyświetlacz widma dźwięku – schemat blokowy.	27
Rys. 24 Wyświetlacz widma dźwięku – schemat połączeń.	29
Rys. 25 Wyświetlacz widma dźwięku – podłączenie MSGEQ7 [37].	30
Rys. 26 Wyświetlacz widma dźwięku – lutowanie diod WS2812B.	31
Rys. 27 Wyświetlacz widma dźwięku – zrealizowany obwód elektryczny.	31
Rys. 28 Wyświetlacz widma dźwięku – gotowe urządzenie.	32
Rys. 29 Inteligentny budynek – konfiguracja programu.	33
Rys. 30 Inteligentny budynek – funkcja startowa.	34
Rys. 31 Inteligentny budynek – główny algorytm programu.	35

Rys. 32 Inteligentny budynek – oprogramowanie alarmu.	36
Rys. 33 Inteligentny budynek – oprogramowanie bramy garażu.....	37
Rys. 34 Inteligentny budynek – fragment oprogramowania wentylatora.	37
Rys. 35 Inteligentny budynek – fragment oprogramowania oświetlenia.	38
Rys. 36 Inteligentny budynek – interfejs użytkownika.....	39
Rys. 37 Inteligentny budynek – komunikacja Bluetooth.	40
Rys. 38 Inteligentny budynek – fragment logiki komunikacji.....	40
Rys. 39 Wyświetlacz widma dźwięku – konfiguracja programu.....	42
Rys. 40 Wyświetlacz widma dźwięku – funkcja startowa.	43
Rys. 41 Wyświetlacz widma dźwięku – główny algorytm.	44
Rys. 42 Wyświetlacz widma dźwięku – ustawienie koloru poszczególnych pikseli.....	45
Rys. 43 Wyświetlacz widma dźwięku – ustawienie wartości koloru diod RGB, na podstawie danych z aplikacji mobilnej.....	46
Rys. 44 Wyświetlacz widma dźwięku – interfejs użytkownika.....	47
Rys. 45 Wyświetlacz widma dźwięku – komunikacja Bluetooth.	48
Rys. 46 Wyświetlacz widma dźwięku – fragment głównej logiki aplikacji.	48

Spis tabel

Tab. 1 Inteligentny budynek – test funkcjonalności urządzenia.	49
Tab. 2 Inteligentny budynek – test komunikacji Bluetooth z urządzeniem.	50
Tab. 3 Wyświetlacz widma dźwięku – test działania diod LED WS2812B w kolumnach.	50
Tab. 4 Wyświetlacz widma dźwięku – test aplikacji mobilnej.....	51
Tab. 5 Wyświetlacz widma dźwięku – test komunikacji Bluetooth z urządzeniem.	51

Bibliografia

- [1] „Montersi,” [Online]. Dostęp: montersi.pl/instrukcje/FGBRS-001-karta.pdf. [Data uzyskania dostępu: 21 Listopad 2021].

- [2] „Serwis,” [Online]. Dostęp: <https://serwis.eu/domy-inteligentne/>. [Data uzyskania dostępu: 21 Listopad 2021].
- [3] „Rynekpierwotny,” [Online]. Dostęp: <https://rynekpierwotny.pl/wiadomosci-mieszkaniowe/inteligentny-dom-co-znaczy-i-ile-kosztuje/9065/>. [Data uzyskania dostępu: 21 Listopad 2021].
- [4] „X-kom,” [Online]. Dostęp: <https://www.x-kom.pl/p/595495-centralka-zestaw-fibaro-home-center-3.html#Specyfikacja>. [Data uzyskania dostępu: 21 Listopad 2021].
- [5] „Allegro,” [Online]. Dostęp: <https://allegro.pl/oferta/fibaro-zestaw-do-inteligentnego-domu-9760262883>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [6] „Cyfra,” [Online]. Dostęp: <https://cyfra.eu/product-pol-4673--Inteligentne-gniazdko-WiFi-Gosund-SP1-Google-Alexa.html>. [Data uzyskania dostępu: 21 Listopad 2021].
- [7] „Epstryk,” [Online]. Dostęp: <https://epstryk.pl/product-pol-47479-Zestaw-6-urzadzen-Fibaro-Starter-Kit-PL.html>. [Data uzyskania dostępu: 21 Listopad 2021].
- [8] „Epstryk,” [Online]. Dostęp: https://epstryk.pl/product-pol-47447-Centrala-Systemu-Fibaro-Home-Center-Lite-FGHCL.html?query_id=1. [Data uzyskania dostępu: 21 Listopad 2021].
- [9] „Epstryk,” [Online]. Dostęp: https://epstryk.pl/product-pol-51262-Czujnik-ruchu-swiatla-temperatury-Fibaro-Motion-Senso.html?query_id=2. [Data uzyskania dostępu: 20 Listopad 2021].
- [10] „Epstryk,” [Online]. Dostęp: https://epstryk.pl/product-pol-47466-Czujnik-dymu-Fibaro-Smoke-Sensor-2-FGSD-002-ZW5.html?query_id=3. [Data uzyskania dostępu: 20 Listopad 2021].
- [11] „Epstryk,” [Online]. Dostęp: https://epstryk.pl/product-pol-47465-Czujnik-zalania-woda-Fibaro-Flood-Sensor-FGFS-001-ZW5.html?query_id=4. [Data uzyskania dostępu: 20 Listopad 2021].
- [12] „Epstryk,” [Online]. Dostęp: https://epstryk.pl/product-pol-47476-Czujnik-zblizeniowy-z-pomiarem-temperatury-FIBARO-Door-Window-Sensor-Bialy-HomeKit-FGBHDW-002-1.html?query_id=7. [Data uzyskania dostępu: 20 Listopad 2021].
- [13] „Epstryk,” [Online]. Dostęp: https://epstryk.pl/product-pol-47478-Inteligentne-gniazdko-z-bolcem-Fibaro-Wall-Plug-E-HomeKit-FGBWHWPE-102.html?query_id=9. [Data uzyskania dostępu: 20 Listopad 2021].

- [14] „Octopart,” [Online]. Dostęp: <https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf>. [Data uzyskania dostępu: 21 Listopad 2021].
- [15] A. V. Chavdarov, „The working principle of an Arduino,” w *11th International Conference on Electronics, Computer and Computation (ICECCO)*, Abuja, Nigeria, 2014.
- [16] „Elfadistelec,” [Online]. Dostęp: <https://www.elfadistelec.pl/pl/plytka-mikrokontrolera-uno-arduino-a000066/p/11038919>. [Data uzyskania dostępu: 20 Grudnia 2021].
- [17] „Quora,” [Online]. Dostęp: <https://www.quora.com/What-is-the-difference-between-the-Arduino-Nano-and-the-Arduino-Uno>. [Data uzyskania dostępu: 19 Grudnia 2021].
- [18] „Conrad,” [Online]. Dostęp: <https://www.conrad.pl/p/plytka-nano-arduino-a000005-nano-1172623>. [Data uzyskania dostępu: 19 Grudzień 2021].
- [19] „Nettigo,” [Online]. Dostęp: <https://nettigo.pl/products/serwomechanizm-feetech-fs90r-analogowe-360-praca-ciagla-4-8-6v>. [Data uzyskania dostępu: 20 Grudzień 2021].
- [20] „Pololu,” [Online]. Dostęp: <https://www.pololu.com/product/2820>. [Data uzyskania dostępu: 20 Grudzień 2021].
- [21] „Wikipedia,” [Online]. Dostęp: [https://pl.wikipedia.org/wiki/PIR_\(czujnik_ruchu\)](https://pl.wikipedia.org/wiki/PIR_(czujnik_ruchu)). [Data uzyskania dostępu: 20 Grudzień 2021].
- [22] „Botland,” [Online]. Dostęp: <https://botland.com.pl/czujniki-ruchu/1655-czujnik-ruchu-pir-hc-sr501-zielony-5903351241359.html>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [23] „Components101,” [Online]. Dostęp: <https://components101.com/sensors/hc-sr501-pir-sensor>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [24] „Arduino,” [Online]. Dostęp: <https://create.arduino.cc/projecthub/electropeak/getting-started-with-hc-05-bluetooth-module-arduino-e0ca81>. [Data uzyskania dostępu: 20 Grudzień 2021].
- [25] „Geeksforgeeks,” [Online]. Dostęp: <https://www.geeksforgeeks.org/all-about-hc-05-bluetooth-module-connection-with-android/>. [Data uzyskania dostępu: 20 Grudzień 2021].
- [26] „Ampere-electronics,” [Online]. Dostęp: <https://ampere-electronics.com/p/hc-05-bluetooth-module-serial-ttl-3/>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [27] „Mercadolibre,” [Online]. Dostęp: https://articulo.mercadolibre.com.ar/MLA-807053524-modulo-bluetooth-hc05-hc-05-maestro-esclavo-uart-arduino-_JM. [Data uzyskania dostępu: 20 Grudzień 2021].

- [28] „Etechnophiles,” [Online]. Dostęp: <https://www.etechnophiles.com/hc-05-pinout-specifications-datasheet/>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [29] „Addressablestrip,” [Online]. Dostęp: <http://pl.addressablestrip.com/info/ws2801-vs-ws2811-vs-ws2812-digital-pixel-led-s-38177117>. [Data uzyskania dostępu: 20 Grudzień 2021].
- [30] „Botland,” [Online]. Dostęp: <https://botland.com.pl/content/218-adresowane-paski-led-rgb-ws2811b-vs-ws2813-czym-sie-roznia>. [Data uzyskania dostępu: 20 Grudzień 2021].
- [31] „Adafruit,” [Online]. Dostęp: <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [32] „Amazon,” [Online]. Dostęp: <https://www.amazon.com/BTF-LIGHTING-Flexible-Individually-Addressable-Non-waterproof/dp/B01CDTEJBG>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [33] „Arrow,” [Online]. Dostęp: <https://www.arrow.com/en/research-and-events/articles/protocol-for-the-ws2812b-programmable-led>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [34] „Seedstudio,” [Online]. Dostęp: <https://www.seedstudio.com/blog/2020/09/30/introduction-of-msgeq7/>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [35] H. Paul i H. Winfield, w *Sztuka elektroniki cz. 1.*, New York, Wydawnictwa Komunikacji i Łączności WKŁ, 2018.
- [36] „Sound-au,” [Online]. Dostęp: <https://sound-au.com/appnotes/an014.htm>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [37] „Jarzebski,” [Online]. Dostęp: <http://www.jarzebski.pl/arduino/komponenty/7-kanalowy-equalizer-msgeq7.html>. [Data uzyskania dostępu: 21 Grudzień 2021].
- [38] „Dfrobot,” [Online]. Dostęp: https://wiki.dfrobot.com/Fan_Module_DFR0332. [Data uzyskania dostępu: 21 Grudzień 2021].
- [39] „Gotronic,” [Online]. Dostęp: <https://www.gotronic.fr/art-ventilateur-gravity-dfr0332-28257.htm>. [Data uzyskania dostępu: 22 Grudzień 2021].
- [40] „Allegro,” [Online]. Dostęp: <https://allegro.pl/oferta/przetwornica-step-down-z-24v-12v-do-5v-dc-5a-25w-9694487542>. [Data uzyskania dostępu: 22 Grudzień 2021].

- [41] „Reddit,” [Online]. Dostęp: https://www.reddit.com/r/arduino/comments/9kwlb/beginner_why_does_my_hc_05_blue_tooth_module_enter/. [Data uzyskania dostępu: 22 Grudzień 2021].
- [42] „Fonewalls,” [Online]. Dostęp: <https://www.fonewalls.com/1080x2340-wallpapers/1080x2340-background-hd-wallpaper-036/>. [Data uzyskania dostępu: 22 Grudzień 2021].
- [43] „Flaticon,” [Online]. Dostęp: <https://www.flaticon.com/>. [Data uzyskania dostępu: 22 Grudzień 2021].
- [44] „Google,” [Online]. Dostęp: <https://play.google.com/store/apps/details?id=com.dutchmatic.patone&hl=pl&gl=US>. [Data uzyskania dostępu: 22 Grudzień 2021].