

**PROJEKT INŻYNIERSKI**

„Osobista skrytka pocztowa”

**Henryk Radziewicz**

**Nr albumu 295484**

**Kierunek:** Elektronika i Telekomunikacja

**PROWADZĄCY PRACĘ**

**Dr inż. Jarosław Wrotniak**

**KATEDRA Systemów Cyfrowych**

**Wydział Automatyki, Elektroniki i Informatyki**

**GLIWICE 2024**

**Streszczenie wraz ze słowami kluczowymi**

**Tytuł pracy:**

Osobista skrytka pocztowa

**Streszczenie:**

Celem pracy inżynierskiej jest stworzenie prototypu osobistej skrytki pocztowej. Prototyp jest sterowany przez specjalnie opracowany program dla mikrokontrolera STM32, który zarządza obsługą peryferii urządzenia. Taka architektura systemu umożliwia komunikację z użytkownikami oraz kontrolę nad mechanizmem zabezpieczającym skrytkę.

**Słowa kluczowe:**

STM32, mikrokontroler, elektronika, skrytka pocztowa, SMS, Bluetooth

**Thesis title:**

Personal PO (Post Office) Box

**Abstract:**

The aim of the engineering thesis is to create a prototype of a personal PO box. The prototype is controlled by a specially developed program written for STM32 microcontroller, which manages the handling of the device’s peripherals. Such a system architecture allows communication with users and control over the PO box’s security mechanism.

**Keywords:**

STM32, microcontroller, electronics, PO box, SMS, Bluetooth

Spis treści

[Rozdział 1 Wstęp 1](#_Toc98759118)

[Rozdział 2 Analiza tematu 3](#_Toc98759119)

[Rozdział 3 Specyfikacja projektu 5](#_Toc98759120)

[Rozdział 4 Specyfikacja zewnętrza - Konstrukcja urządzenia 7](#_Toc98759121)

[Rozdział 5 Specyfikacja wewnętrzna - Oprogramowanie 9](#_Toc98759122)

[Rozdział 6 Weryfikacja i walidacja 11](#_Toc98759123)

[Rozdział 7 Podsumowanie i wnioski 13](#_Toc98759124)

[Bibliografia 15](#_Toc98759125)

[Spis skrótów i symboli 19](#_Toc98759126)

[Źródła 20](#_Toc98759127)

[Lista dodatkowych plików, uzupełniających tekst pracy 21](#_Toc98759128)

[Spis rysunków 22](#_Toc98759129)

[Spis tablic 23](#_Toc98759130)

# Rozdział 1 Wstęp

Od początku XXI wieku można zaobserwować dynamiczny wzrost rynku handlu elektronicznego (ang. *e-commerce*), co skutkuje znacznym rozwojem w sektorze usług kurierskich. Wynika to ze zwiększonego ruchu przesyłek dostarczanych do konsumentów, którzy zamawiają coraz więcej produktów z różnych sklepów internetowych. Rozwój rynku e-commerce umożliwił również małym przedsiębiorcom oferowanie swoich produktów wyłącznie poprzez wysyłkę bezpośrednio do zainteresowanych osób, bez konieczności otwierania sklepu stacjonarnego. Obydwie wymienione grupy wykorzystują do swoich celów usługi kurierskie, które przez długi czas ograniczały się jedynie do dostawy bądź odbioru bezpośrednio pod podany adres lub do punktu kurierskiego znajdującego się często w niewygodnej dla klienta lokalizacji. Dostawy przesyłek pod podany adres również posiadają kilka wad. Głównym problemem jest to, że kurierzy pracują w godzinach zbliżonych do standardowych godzin pracy większości ludzi. W związku z tym, w przypadku próby doręczenia paczki przez kuriera podczas nieobecności odbiorcy mogą wystąpić trzy opcje. Pierwsza to powiadomienie odbiorcy o nieudanej próbie dostarczenia przesyłki i jej odesłanie do punktu kurierskiego, skąd zostanie ponownie podjęta próba dostarczenia następnego dnia. Druga to pozostawienie paczki u sąsiada adresata, co może wiązać się z ryzykiem, iż osoba trzecia nie została poinformowana o takiej możliwości lub kwestiami braku zaufania wobec zastępczego odbiorcy. Ostatnia opcja często występująca szczególnie w krajach anglosaskich to pozostawienie przesyłki przed drzwiami odbiorcy, co niesie ze sobą ryzyko kradzieży przez tak zwanych „Porch Pirate”. Poruszony problem został zauważony przez firmy e-commerce oraz firmy kurierskie. W odpowiedzi na to, podjęto inicjatywę mającą na celu znalezienie rozwiązania, polegające na zwiększeniu liczby punktów obsługi klienta. Są one rozmieszczone w łatwo dostępnych lokalizacjach, tworząc gęstą sieć na obszarach miejskich. Jednakże te punkty często powstają w już istniejących sklepach, które mają ograniczone godziny otwarcia. Ponadto przesyłki są wydawane przez kasjerów, którzy jednocześnie obsługują również klientów sklepowych. Problemy te są podobne do tych, które pojawiają się, gdy kurier nie zastaje odbiorcy pod wyznaczonym adresem, co często przypomina pierwszą opcję wcześniej opisanej sytuacji, a dodatkowo dochodzi do zbędnego przeznaczania czasu na oczekiwanie w celu odebrania lub nadania paczki.

Wymienione wyzwania dotyczące odbierania i nadawania przesyłek nie tylko mieszczą się w dziedzinie związanej z zagrożeniami związanymi z bezpieczeństwem i wygodą, ale obejmuje również kwestię marnotrawstwa czasu, co jest kluczowe przy korzystaniu z usług kurierskich.

Celem pracy inżynierskiej jest stworzenie prototypu urządzenia obsługującego osobistą skrytkę pocztową, umożliwiającą bezkontaktowe odbieranie i nadawanie paczek. Urządzenie ma być sterowane przez użytkownika za pomocą telefonu połączonego poprzez Bluetooth oraz mieć funkcję wysyłania wiadomości SMS (ang. *Short Message Service*) z kodem dostępu do skrytki dla kuriera. Zewnętrzna część prototypu składa się z zamka blokującego dostęp do wnętrza skrytki przez osoby postronne oraz klawiatury membranowej z wyświetlaczem LCD (ang. *Liquid-Crystal Display*), która umożliwia wprowadzenie wysłanego kodu dostępu dla kuriera.

Zakres pracy obejmuje skompletowanie i opisanie niezbędnych modułów komunikacyjnych takich jak Bluetooth oraz GSM (ang. *Global System for Mobile Communications*) oraz peryferii zewnętrznych, które będą obsługiwane przez mikrokontroler STM32 w ramach zaprojekowanego urządzenia. Z uwagi na to, że projekt wymaga zakończenia poprzez uruchomienie prototypu, zakres pracy obejmuje również umieszczenie urządzenia przy obiekcie, który będzie służył za skrytkę pocztową.

Praca inżynierska składa się z siedmiu rozdziałów, z których każdy skupia się na innym aspekcie projektu. Wstęp wprowadza temat, przedstawiając cel i zakres pracy oraz osadzając problem w danej dziedzinie. Drugi rozdział to szczegółowa analiza problemu dodatkowo prezentująca istniejące rozwiązania rynkowe. Trzeci rozdział to specyfikacja projektu, zawierająca uzasadnienie dokonanych wyborów oraz opis wykorzystanych narzędzi. Czwarty rozdział omawia specyfikację zewnętrzną urządzenia, w tym jego fizyczną konstrukcję i schemat działania. Piąty rozdział koncentruje się na specyfikacji wewnętrznej, czyli na aspektach programistycznych urządzenia. Szósty rozdział obejmuje weryfikację i walidację, opisuje procedurę testowania urządzenia z napotkanymi problemami i opracowanymi rozwiązaniami. Ostatni rozdział - Podsumowanie i wnioski, zawiera przegląd spełnienia założeń projektu, dostępnych rozwiązań postawionych problemów oraz możliwego kierunku rozwoju urządzenia.

# Rozdział 2 Analiza tematu

## Analiza problemu kradzieży oraz podejmowane próby rozwiązania problemu

Zjawisko kradzieży paczek pozostawionych przed drzwiami domów staje się coraz bardziej powszechne w krajach anglojęzycznych, w tym Stanach Zjednoczonych, Kanadzie, Australii czy Wielkiej Brytanii. Szacuje się, że w samych Stanach Zjednoczonych codziennie dochodzi do kradzieży około 1.7 miliona przesyłek, co generuje straty w wysokości 25 milionów dolarów w wartości towarów i usług [1]. Średni koszt zastąpienia przedmiotu skradzionego z paczki wynosi 109 dolarów obciążając zarówno sprzedawców, jak i przewoźników [1]. Biorąc pod uwagę, że ponad połowa właścicieli domów w USA otrzymuje dostawy co najmniej raz w tygodniu, jest to szczególnie dotkliwe nie tylko dla firm oferujących produkty lub usługi kurierskie, ale również dla samych klientów. Często ofiarą kradzieży przez tzw. „Porch Pirates” stają są przesyłki, które nie zawierają wartościowych nowych przedmiotów zakupionych bezpośrednio ze sklepu internetowego, lecz przedmioty znacznie trudniejsze do zrekompensowania przez przewoźnika. W tej kategorii znajdują się, między innymi, leki, pisma urzędowe, w tym dokumenty zawierające renty oraz emerytury, a także rzeczy o wartości sentymentalnej.

Metody działania złodziei paczek są różnorodne, począwszy od bezpośredniego zawłaszczenia pozostawionych przesyłek, aż po bardziej skomplikowane sposoby. Najczęściej złodzieje podszywają się za kurierów, ubierając podobne uniformy lub śledzą ich podczas dostaw [2]. Innym planem działania złodziei jest przeprowadzanie zwiadu w celu ustalenia godzin nieobecności mieszkańców. Kiedy kurierzy zostawiają paczki pod drzwiami w czasie pobytu adresatów w pracy lub szkole, ryzyko zawłaszczenia przesyłki przez osobę niepowołaną znacząco wzrasta. Początkowo, w odpowiedzi na występujący problem, klienci sami próbowali znaleźć rozwiązanie. Jednym ze sposobów było podawanie przy składaniu zamówień adresu miejsca pracy, pod którym przebywali podczas nieobecności w domu, aby paczka mogła być bezpiecznie odebrana. Niestety, nie wszyscy mają możliwość skorzystania z tej opcji, na przykład ze względu na pracę zdalną lub bycie na emeryturze. Jednakże, kiedy problem kradzieży paczek został zauważony zarówno przez firmy e-commerce, jak i przewoźników, podjęto próby wprowadzenia rozwiązań, mających na celu ograniczenie tego zjawiska. Został rozbudowany i upowszechniony system śledzenia przesyłek. Wprowadzono również możliwość pozostawienia instrukcji dotyczących umieszczenia paczki w bezpiecznym miejscu. Ponadto, firmy kurierskie rozpoczęły szkolenia pracowników w zakresie bezpiecznego ukrywania przesyłek. Dodatkowo, kwestię rozwiązania istniejącego problemu podejmują zewnętrzne firmy, które liczą na generowanie dochodów z oferowanych przez siebie produktów lub usług. Przykładem takiego podejścia, wspominanego we wstępie, jest udostępnianie sklepów do wydawania i przyjmowania paczek kurierskich. Innym rozwiązaniem oferowanym przez przedsiębiorstwa specjalizujące się w elektronice lub automatyce domowej są domofony wyposażone w kamery. Te urządzenia monitorują obszar przed drzwiami i powiadamiają użytkowników o ewentualnych niepokojących sytuacjach.

Zastosowane metody przyczyniły się do zmniejszenia ryzyka kradzieży paczek, jednakże nie doprowadziły do całkowitego wyeliminowania problemu. System śledzenia przesyłek dostarcza jedynie informacje o bieżącym statusie paczki, które często są zapełnione nieistotnymi szczegółami dotyczącymi przyjęcia lub wysłania przesyłki z centrum logistycznego przewoźnika. Po dostarczeniu, paczka pozostaje bez nadzoru. Przekazywanie instrukcji dotyczących bezpiecznego umieszczenia przesyłki oraz szkolenia kurierów w tym zakresie, z czasem tracą swoją skuteczność. Wynika to z faktu, że złodzieje adaptują się, zapamiętując miejsca uznawane za bezpieczne oraz taktyki stosowane przez kurierów. Rozwiązaniem, które umożliwia na lepszą kontrolę sytuacji przed domem są wspominane wcześniej domofony z wbudowanymi kamerami. Pozwalają one mieszkańcom na szybszą rekcję w przypadku nieprzyjemnych incydentów oraz umożliwiają nagranie sprawcy przywłaszczenia dostarczonej przesyłki. Sukces domofonów z funkcją podglądu terenu wokół domu odzwierciedla się w wynikach sprzedażowych, które wzrosły z 100 000 sztuk w 2014 roku do 1.2 miliona sztuk w 2019 roku [1]. Pomimo osiągnięcia sukcesu w zakresie poprawy przydomowego bezpieczeństwa, problem kradzieży dalej występuje. Na podstawie przeprowadzonych badań ankietowych szacuje się, że średnio co piąty obywatel Stanów Zjednoczonych padł ofiarą kradzieży paczki w latach 2020 - 2023 [3]. Firmy przewozowe oraz e-commerce kontynuowały poszukiwania rozwiązania, które umożliwiłoby bezpieczne pozostawienie przesyłki przy jednoczesnym zachowaniu niskich kosztów dostawy. W rezultacie tych działań powstały publiczne, samoobsługowe punkty odbioru i nadawania paczek, znane jako automaty paczkowe.

## Znane rozwiązania

Jednym z najbardziej znanych automatów paczkowych jest „Paczkomat®” firmy InPost zaprezentowany na Rysunku 1. Jest to największa firma kurierska w Polsce pod względem oferowanych automatów paczkowych. Udostępnia ponad 20 000 samoobsługowych punktów odbioru i nadawania, rozmieszczonych na terenie całego kraju [4]. Ponieważ, odległości pomiędzy sąsiednimi miejscowościami w krajach europejskich są znacznie mniejsze niż w Stanach Zjednoczonych, sieć samoobsługowych automatów paczkowych umożliwiających bezpieczny odbiór niezależnie od pory dnia mocno ograniczyły występowanie kradzieży paczek, gdyż dla klientów zamówienie paczki do „Paczkomatu” stało się wygodniejsze. Pomimo bardziej dogodnych lokalizacji automatów paczkowych oraz możliwości ekspresowego skorzystania z usługi, użytkownicy często zmuszeni są do nadkładania drogi, aby odebrać przesyłkę.

Obraz zawierający Prostokąt, budynek, design

Opis wygenerowany automatycznie

Rysunek 1. „Paczkomat®” firmy InPost

Podobne rozwiązania zostały wprowadzone przez firmę Amazon, będącą największą amerykańską platformą e-commerce. Ich produkt, „Amazon Hub Locker” widoczny na Rysunku 2., jest dostępny w niemal 17 tysiącach lokalizacji na terenie Stanów Zjednoczonych [5]. Jednak, gdy porównuje się skalę dostępnych automatów firmy InPost w Polsce, kraju o powierzchni 30 razy mniejszej i populacji 10 razy mniejszej niż Stany Zjednoczone, rozwiązania Amazonu wydają się mieć mniejszy wpływ w porównaniu z skalą działania polskiej firmy. Automaty paczkowe firmy Amazon są zlokalizowane głównie w dużych miastach, co może prowadzić do pominięcia obszarów wiejskich i mniejszych miasteczek.

Rysunek 2. Automat paczkowy "Amazon Hub Locker" firmy Amazon

Obraz zawierający Szafa kartotekowa, ściana, w pomieszczeniu, tekst

Opis wygenerowany automatycznie

Kwestię poprawy wygody, bezpieczeństwa i oszczędności czasu związaną z odbiorem paczek podjęły również mniejsze prywatne firmy. Alternatywą dla dużych automatów paczkowych mogą być kompaktowe skrzynki do odbioru paczek, instalowane w obrębie domu i mieszczące pojedynczą przesyłkę. Przykładem takiego rozwiązania jest produkt „Paczkoport S-1N SZ”, prezentowany na Rysunku 3., oferowany przez firmę Paczkoport. Skrzynki na paczki dla osób prywatnych, umożliwiające pozostawienie przesyłek przez kuriera, opierają się głównie na zabezpieczeniach mechanicznych, wykorzystując zapadanie blokujące dostęp do komory na paczki przy otwartych drzwiczkach do wrzucenia przesyłki, co zapewnia bezpiecznie dostarczenie paczki pod dom adresata, przy jednoczesnym utrudnieniu osobom niepowołanym wyciągnięcia przesyłki ze skrzynki.

Obraz zawierający design

Opis wygenerowany automatycznie przy niskim poziomie pewności

Rysunek 3. "Paczkoport S-1N SZ" firmy Paczkoport

Zaprezentowany model „Paczkoportu” posiada również, zamek odblokowywany kluczem lub wprowadzonym kodem. Umożliwia to odbiór przesyłki przez kuriera przy użyciu kodu dostępu wcześniej podanego przez nadawcę. Szyfrowy zamek blokujący dostęp do skrytki jest wykorzystywany na szeroką skalę przez inne firmy oferujące podobne rozwiązania. Jednak ciągłe używanie tego samego kodu dostępu może stanowić wrażliwy punkt omawianego rozwiązania.

# Rozdział 3 Specyfikacja projektu

## 3.1 Założenia konstrukcyjne prototypu

Po przeprowadzonej analizie postawionego problemu i dostępnych na rynku rozwiązań, możliwe jest określenie wymagań, którym konstruowany prototyp powinien sprostać. Ze względu na temat pracy, dotyczący osobistej skrytki pocztowej, prototyp powinien charakteryzować się kompaktowymi rozmiarami i zdolnością pomieszczenia jednej średniej wielkości paczki. Dodatkowo, zaleca się, aby poprawił kwestię bezpieczeństwa poprzez ograniczanie liczby ruchomych elementów oraz wprowadzenie zmiennego kodu dostępu, który odblokowuje zamek. Najlepiej, żeby obsługa prototypu była intuicyjna i w pewnym stopniu spójna z rozwiązaniami stosowanymi w automatach paczkowych. Biorąc pod uwagę podane wymagania, prototyp będzie składał się z wymienionych elementów:

* Płytki developerskiej Nucleo-64 z mikrokontrolerem STM32
* Modułu GSM z anteną poprawiającą zasięg sieci komórkowej
* Modułu Bluetooth
* Serwomechanizmu obsługującego zamek skrytki
* Czujnika magnetycznego
* Membranowej klawiatury matrycowej 4x4
* Wyświetlacza alfa-numerycznego LCD 2x16

W celu osiągnięcia optymalnych rezultatów konstruowanego prototypu, skupiono się na modularności umożliwiającej na łatwą wymianę poszczególnych elementów. Płytka developerska Nucleo-64 posiada wyprowadzenia typu „goldpin” o rastrze 2,54 mm, nazywane przez producenta płytki jako „Morpho extension” [7]. Dodatkowo Nucleo-64 posiada wbudowany programator-debugger, który umożliwia łatwe wgrywanie oraz debugowanie napisanego kodu. Wybór modułu GSM został podyktowany faktem, że zasięg sieci komórkowej 2G, która pracuje na niższych częstotliwościach jest lepsza w miejscach trudno dostępnych niż w przypadku nowszych technologii 3G/4G, oraz jest wykorzystywany tylko do wysyłania wiadomości SMS. Moduł Bluetooth służy do komunikacji z telefonem użytkownika, nadającym lub odbierającym przesyłki. Z założenia prototyp ma być urządzeniem „standalone” (ang. *samodzielny*), więc komunikacja bliskiego zasięgu jest wystarczająca. Serwomechanizm w połączeniu z czujnikiem magnetycznym prezentuje model przykładowego zamka zabezpieczającego wnętrze skrytki. Sam mechanizm w prototypie nie jest wystarczający do szczelnego zablokowania dostępu przy użyciu siły, ale wystarcza do wizualizacji mechanizmu i testowania opracowanego kodu. Membranowa klawiatura matrycowa o wymiarach 4x4 przyciski jest przeznaczona do wpisywania kodu dostępu do skrytki, wysłanego w wiadomości SMS oraz zatwierdzenia dostarczenia lub odebrania przesyłki. Zastosowanie fizycznej klawiatury eliminuje konieczność łączenia się kuriera zewnętrznym urządzeniem np. telefonem ze skrytką co przyspiesza interakcje z prototypem. Wyświetlacz alfa-numeryczny LCD z podświetlaniem umożliwia przejrzyste wyświetlanie komunikatów dotyczących stanu urządzenia użytkownikom.

## 3.2 Środowisko pracy

W procesie tworzenia prototypu kluczową role odgrywa opracowane oprogramowanie przeznaczone dla mikrokontrolera do obsługi przedstawionych wcześniej komponentów. W tym celu wykorzystano środowisko programistyczne STM32CubeIDE, które jest udostępniane bezpłatnie przez samą firmę STM (*STMicroelectronics*) [8]. STM32CubeIDE jest rozbudowanym programem, który umożliwia łatwe tworzenie projektów korzystających z bibliotek HAL (ang. *Hardware Abstraction Layer*) zaprojektowanych przez firmę STM. Biblioteki te pozwalają na łatwiejszą obsługę peryferii sprzętowych mikrokontrolera bez konieczności samodzielnego pisania bibliotek opartych na rejestrach. Dodatkowo, STM32CubeIDE posiada interfejs graficzny z trzema perspektywami: „Device Configuration Tool”, „C/C++” oraz „Debug”. „Device Configuration Tool”, dostępny również jako STM32CubeMX, służy do generowania kodu na podstawie wybranych opcji, włączających interfejsy komunikacyjne oraz zadeklarowanych pinów wejścia/wyjścia. Perspektywa „C/C++” jest przeznaczona do pisania kodu w plikach źródłowych oraz bibliotekach, natomiast „Debug” służy do debugowania kodu uruchomionego na mikrokontrolerze w czasie rzeczywistym.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Rysunek 3.1 Perspektywa „Device Configuration Tool” programu STM32CubeIDE

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Rysunek 3.2 Perspektywa „C/C++” programu STM32CubeIDE

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Rysunek 3.3 Perspektywa „Debug” programu STM32CubeIDE

W ramach debugowania został również użyty program Tera Term, który umożliwia komunikację komputera z mikrokontrolerem przez UART (ang. *Universal Asynchronous Receiver-Transmitter*). Program Tera Term służy do kontrolowania komunikatów wysyłanych przez STM32 w celu wychwycenia ewentualnych problemów ze strony modułów Bluetooth oraz GSM, a także umożliwia komunikację z tymi modułami.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Rysunek 3.4 Program Tera Term

Do komunikacji urządzenia z telefonem przez interfejs Bluetooth wykorzystuje się aplikację nRF Connect stworzoną przez Nordic Semiconductor [9]. Aplikacja umożliwia debugowanie urządzeń Bluetooth oraz pobieranie i przesyłanie danych. Dodatkowo, aplikacja daje wgląd w usługi oraz charakterystyki udostępnianych przez urządzenia.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Rysunek 3.5 Aplikacja nRF Connect

# Rozdział 4 Specyfikacja zewnętrzna – Konstrukcja urządzenia

## 4.1 Budowa zewnętrzna prototypu

Prototyp powstały w ramach projektu inżynierskiego składa się z modułów przedstawionych w rozdziale 3, podłączonych zgodnie ze schematem przedstawionym na Rysunku 4.1. Dodatkowo w Tabeli 4.1 zostały wypisane wszystkie użyte komponenty użyte w konstrukcji prototypu.

Obraz zawierający tekst, diagram, Plan

Opis wygenerowany automatycznie

Rysunek 4.1 Schemat podłączenia modułów prototypu

Tabela 4.1 Lista użytych komponentów do stworzenia prototypu

|  |  |
| --- | --- |
| Ilość | Nazwa modułu |
| 1 | Płytka deweloperska Nucleo-L476RG z mikrokontrolerem STM32L476RGT6 |
| 1 | Moduł 2G „u-GSM shield” z modemem Quectel M95FA |
| 1 | Antena GSM SMA z przewodem SMA – u.FL |
| 1 | Moduł Bluetooth 4.0 BLE HM-10/AT-09 |
| 1 | Wyświetlacz LCD 2x16 znaków ze sterownikiem HD44780 |
| 1 | Konwerter I2C oparty na PCF8574 |
| 1 | Klawiatura membranowa matrycowa 4x4 |
| 1 | Serwomechanizm Serwo TowerPro SG-90 mikro |
| 1 | Czujnik magnetyczny - kontaktron |

Głównym elementem prototypu jest Nucleo-L476RG, który poza obsługą podłączonych modułów, dostarcza również zasilanie poprzez złącze USB (ang. *Universal Serial Bus*). Mikrokontroler STM32L476RGT6, znajdujący się na płytce deweloperskiej, komunikuje się za pomocą UART z modułem GSM i Bluetooth oraz urządzeniem debugująycm poprzez USB. Inną wykorzystywaną formą komunikacją jest I2C (ang. *Inter-Integrated Circuit*), która służy wyłącznie do przesyłania informacji do konwertera PCF8574. Konwerter ten przekształca dane szeregowe na równoległe, które są następnie odczytywane przez sterownik HD44780 wyświetlacza LCD.

## 4.2 Charakterystyka podzespołów

**Nucleo-L476RG** to płytka developerska składająca się z mikrokontrolera STM32L476RGT6 z wyprowadzeniami typu „Morpho extension” oraz zgodnymi z Arduino UNO [7]. Dodatkowo, płytka posiada wbudowany programator-debugger ST-Link/V2-1, który umożliwia na szybkie wgrywanie oraz debugowanie kodu i zasilanie całego urządzenia przez złącze mikro-USB. Wybór mikrokontrolera został podjęty ze względu na dużą wydajność i energooszczędność wykorzystanego 32-bitowego mikroprocesora ARM® Cortex®-M4, a także rozbudowane peryferia sprzętowe zawierające m.in. 6 interfejsów UART, 3 interfejsy I2C, 17 timerów, z których niektóre umożliwiają obsługę PWM (ang. *Pulse-Width Modulation*), oraz sprzętowe wsparcie dla RNG (ang. *Random Number Generator*) [12]. Ponadto Nucleo-L476RG dysponuje 51 pinami GPIO (ang. *General-Purpose Input/Output*) pokazanymi na Rysunku 4.3., które można wykorzystać dla komponentów nie korzystających z dostępnych interfejsów komunikacyjnych.

Obraz zawierający Inżynieria elektroniczna, obwód, elektronika, Komponent elektroniczny

Opis wygenerowany automatycznie

Rysunek 4.2 Nucleo-L476RG [10]

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Rysunek 4.3 Opis wyprowadzeń GPIO płytki Nucleo-L476RG [11]

Przedstawiony na Rysunku 4.3. schemat wyprowadzeń GPIO, ułożony w „Morpho extension” i zgodny z wyprowadzeniami Arduino UNO posłuży do podłączenia peryferii zewnętrznych. Moduł GSM jest podłączony do pinów PA10 (USART1\_RX), PA9 (USART1\_TX) oraz PB5 (MODEM\_ENABLE), a moduł Bluetooth do pinów PC4 (USART3\_TX) i PC5 (USART3\_RX). Wyświetlacz LCD, wykorzystujący komunikację I2C, korzysta z pinu PB7 (I2C1\_SDA) do przesyłania danych oraz PB6 (I2C1\_SCL), który jest linią zegarową. Klawiatura matrycowa jest podłączona do pinów PC6, PC7, PC8, PC9 do których są przyłączone wiersze klawiatury oraz PB12, PB13, PB14, PB15 podłączonych do kolumn matrycy. Przy podłączaniu klawiatury matrycowej ważne było ulokowanie pinów kolumn i wierszy w jednym porcie mikrokontrolera co zapobiega tworzeniu nadmiarowego kodu w strukturze HAL. Serwomechanizm korzysta z pinu PA0 (SERVO\_0), który jest podłączony do licznika obsługującego PWM. Ostatni pin wykorzystany w prototypie jest używany przez czujnik magnetyczny, który jest podłączony do wyprowadzenia PB11 (MAG\_SWITCH). Dodatkowo, moduły wymagające zasilania są podłączone do pinu 5V zasilanego bezpośrednio z portu mikro-USB płytki developerskiej, oraz pinu GND (ang. *ground*).

**Moduł 2G „u-GSM shield”** wraz z **Anteną GSM SMA** podłączoną do modułu GSM przez przewód SMA (ang. *SubMiniature version A*) - u.FL, tworzą system komunikacyjny GSM do przesyłania wiadomości SMS. Moduł jest oparty na modemie GSM Quectel M95FA, posiada wbudowany slot na kartę SIM (ang. *Subscriber Identity Module*), konwerter USB-UART oraz układ ładujący akumulator Li-Po (*litowo-polimerowy*) lub akumulator Li-Ion (*litowo- jonowy*). Może również być zastosowany superkondensator zamiast akumulatora. Moduł GSM posiada 10 wyprowadzeń, jednak w projekcie wykorzystano tylko pięć pinów. Piny RX (ang. *receiver*) oraz TX (ang. *transmitter*) służą do komunikacji przez interfejs UART korzystając z komend AT, przy czym należy zwrócić uwagę, że RX/TX nie oznaczają odbiornika/transmitera modułu, lecz mikrokontrolera, prawdziwa funkcjonalność tych pinów jest odwrócona. Dodatkowo zaimplementowano obsługę pinu Enable, aby umożliwić uruchomienie modemu z niewielkim opóźnieniem po włączeniu urządzenia. Oraz wykorzystano piny Vin i GND jako zasilania modemu.

Obraz zawierający tekst, Element obwodu, Komponent elektroniczny, Pasywny element obwodu

Opis wygenerowany automatycznie

Rysunek 4.4 Moduł 2G „u-GSM shield” [13]

Obraz zawierający przewód

Opis wygenerowany automatycznie

Rysunek 4.5 Antena GSM z przewodem SMA – u.FL[14]

**Moduł Bluetooth 4.0 BLE HM-10/AT-09** służy w projekcie do bezprzewodowej komunikacji bliskiego zasięgu. Bazuje na układzie CC2541, posiada także antenę mikropaskową przystosowaną dla częstotliwości Bluetooth. Moduł posiada 6 wyprowadzeń, jednak w projekcie wykorzystane są 4. RX i TX do komunikacji przez UART komendami AT z samym modułem oraz VCC i GND do zasilania. Usługi rozgłaszane przez Bluetooth umożliwiają na komunikację UART, przeprowadzaną bezpośrednio na piny RX/TX, co znacząco ułatwia obsługę modułu Bluetooth [15].

Obraz zawierający Komponent elektroniczny, Element obwodu, Inżynieria elektroniczna, elektronika

Opis wygenerowany automatycznie

Rysunek 4.6 Moduł Bluetooth HM-10 [16]

**Wyświetlacz alfa-numeryczny LCD 2x16 HD44780** razem z **Konwerterem I2C opartym na PCF8574** tworzą układ wyświetlacza sterowanego przez komunikację I2C. Konwerter I2C wymaga tylko czterech pinów: dwóch pinów komunikacyjnych: linii danych SDA (ang. *Serial Data Line*) oraz linii zegarowej SCL (ang. *Serial Clock Line*) oraz dwóch pinów zasilania VCC i GND. Układ PCF8574 to 8-bitowy ekspander, który konwertuje dane szeregowe przesyłane przez I2C na 8 równoległych pinów, a także umożliwia konwersję w przeciwnym kierunku.

Obraz zawierający elektronika, Inżynieria elektroniczna, obwód, Komponent elektroniczny

Opis wygenerowany automatycznie

Rysunek 4.7 Konwerter I2C [17]

Obraz zawierający tekst, diagram, Plan, wykres

Opis wygenerowany automatycznie

Rysunek 4.8 Schemat blokowy układu PCF8574 [18]

Obraz zawierający tekst, Czcionka, linia, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 4.9 Transmisja I2C konwertera PCF8574 [18]

Zaprezentowany schemat blokowy (Rysunek 4.8) oraz przebiegi transmisji I2C (Rysunek 4.9) ilustrują sposób działania konwertera, w którym przesłany szeregowo bajt zmienia stan poszczególnych wyjść P0 - P7. Przykładowo, przesłanie wartości 0x20 (0b0010 0000) ustawia stan wysoki na pinie P5 za pomocą rejestru przesuwnego SIPO (ang. *Serial-In Parallel-Out*). Wejścia A0 – A2 deklarują adres układu. W przypadku wykorzystanego konwertera wartość ta jest stała i ma wynosi 0x27, ponieważ piny A0 – A2 są podłączone do zasilania układu, utrzymując je w stanie wysokim, zgodnie z informacjami przedstawionymi na Rysunku 4.10.

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Rysunek 4.10 Mapa adresowa układu PCF8574 [18]

Obraz zawierający elektronika, Komponent elektroniczny, Element obwodu, Element komputera

Opis wygenerowany automatycznie

Rysunek 4.11 Wyświetlacz alfa-numeryczny LCD 2x16 [20]

Wyświetlacz alfa-numeryczny LCD, zaprezentowany na Rysunku 4.11, umożliwia wyświetlanie 2 wierszy po 16 znaków i jest sterowany przez układ HD44780. Posiada 16 pinów umożliwiających sterowanie w kilku możliwych trybach. W projekcie został wykorzystany tryb 4-bitowy bez odczytu flagi zajętości, to pozwoliło ograniczyć ilość wykorzystywanych pinów do siedmiu: RS – wybór rejestru instrukcji lub danych, EN – Enable, który aktywuje układ HD44780, BL – „Back Light”, czyli podświetlanie wyświetlacza oraz 4-bitową linie danych D4-D7, która służy do przesyłania komend lub adresów znaków przeznaczonych do wyświetlenia, 4-bitowy tryb pracy wymaga przesłanie komendy lub adresu w dwóch półbajtach co jest zaprezentowane na Rysunku 4.12 [19].

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Rysunek 4.12 Transmisja układu HD44780 w trybie 4-bitowym

Obraz zawierający design, kalkulator

Opis wygenerowany automatycznie

Rysunek 4.13 Klawiatura membranowa 4x4 [21]

**Klawiatura membranowa matrycowa 4x4** (Rysunek 4.13)umożliwia bezpośrednią interakcję z urządzeniem i stanowi zabezpieczeniem oddzielające możliwość wykonania akcji, które obsługuje prototyp przez interfejs Bluetooth. Budowa klawiatury membranowej jest bardzo prosta, ponieważ składa się z kilku warstw. Pierwsza elastyczna warstwa, widoczna z zewnątrz posiada nadrukowane znaki, pełni funkcję ochronną oraz utrzymuje kolejne warstwy w odpowiednich miejscach. Druga warstwa posiada naniesione ścieżki kolumn, a trzecia ścieżki wierszy, dodatkowo pomiędzy drugą a trzecią warstwą znajduje się separator, który zapobiega przypadkowym naciśnięciom. Podczas naciskania punkt na drugiej i trzeciej warstwie zwiera się, łącząc konkretną kolumnę z konkretnym wierszem, co pozwala na odczytanie konkretnie wciśniętego przycisku. Klawiatura matrycowa jest czysto fizycznym elementem, którego działanie dopiero implementuje się w oprogramowaniu.

Obraz zawierający tekst, diagram, linia, Wykres

Opis wygenerowany automatycznie

Rysunek 4.14 Schemat klawiatury matrycowej

**Serwomechanizm TowerPro SG-90 mikro** jest elementem modelującym zamek w konstruowanym prototypie. Jego budowa wewnętrzna serwomechanizmu opiera się na silniku prądu stałego, który napędza przekładnie kół zębatych, oraz jest sterowany specjalnym układem [22]. Wszystkie wymienione elementy serwomechanizmu są umieszczone w jednej obudowie, co zostało zaprezentowane na Rysunku 4.15.

Obraz zawierający plastik

Opis wygenerowany automatycznie

Rysunek 4.15 Serwomechanizm TowerPro SG-90 mikro [23]

Pozycja wychylenia przekładni jest ustawiana przez sygnał PWM podawany na pin wejściowy serwomechanizmu. W zależności od szerokości impulsu, których czas trwania wynosi od 1 ms do 2 ms, przekładnia serwomechanizmu jest wychylana w lewo (dla krótszego impulsu) lub w prawo (dla dłuższego). Do poprawnego działania, układ sterujący serwomechanizmu wymaga powtarzania impulsu sterującego z częstotliwością 50 Hz, co daje okres powtarzania co 20 ms [22].

Obraz zawierający zrzut ekranu, tekst, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Rysunek 4.16 Przykładowy przebieg PWM sterującego serwomechanizmem [24]

**Czujnik magnetyczny – kontaktron** służy do kontroli zamknięcia drzwiczek prototypu. Jego budowa jest bardzo prosta i opiera się na łączniku elektrycznym, który stanowią dwie cienkie blaszki przewodnika umieszczone w szklanej tubie [25]. Pod wpływem pola magnetycznego blaszki kontaktronu wychylają się, doprowadzając do zetknięcia i zamknięcia obwodu. Wykorzystany w prototypie kontaktron został umieszczony w plastikowej osłonce.

Obraz zawierający lampa elektronowa

Opis wygenerowany automatycznie

Rysunek 4.17 Kontaktron [26]

## 4.3 Zasilanie prototypu

Zapewnienie odpowiedniego zasilania dla wszystkich modułów wykorzystywanych w projekcie jest kluczowe, aby uniknąć nieprawidłowego funkcjonowania. Prototyp jest zasilany poprzez złącze USB płytki developerskiej Nucleo-L476RG. Linia zasilająca USB jest podłączona do 5V stabilizatora LD1117S50TR znajdującego się na Nucleo-L476RG, gdzie stabilizowane napięcie jest podawane na wyprowadzenia 5V pinów „Morpho extension” [27]. Maksymalny prąd wyjściowy dla stabilizatora LD1117S50TR wynosi 1.3 A, co powinno wystarczyć dla bazowego zasilania całego prototypu [28]. Napięcie zasilania dla mikrokontrolera STM32L476RGT6 jest dodatkowo obniżane do 3.3V przez dodatkowy regulator LDO (ang. *Low-Dropout Regulator*) LD39050PU33R [27]. Obniżone napięcie 3.3V jest stosowane przez mikrokontroler na pinach wyjścia/wejścia do komunikacji z dodatkowymi peryferiami. Moduł GSM, Bluetooth, wyświetlacz LCD z konwerterem I2C oraz serwomechanizm jest zasilany napięciem 5V.

Tabela 4.2 Zapotrzebowanie prądowe prototypu

|  |  |
| --- | --- |
| Moduł | Pobór prądu [mA] |
| Mikrokontroler STM32L476RGT6 | 8.96 |
| Moduł GSM z modemem Quectel M95FA | 13 (podczas działania występują skoki prądowe do 1.6 A) |
| Moduł Bluetooth - CC2541 | 20.2 |
| Konwerter I2C - PCF8574 | 0.1 |
| Wyświetlacz LCD – HD44780 | 42.5 (razem z podświetleniem wyświetlacza) |
| Serwomechanizm SG-90 | 10 (podczas obrotu występuje skok do 250 mA) |
| Suma: | 94.76 mA |

Prototyp składa się z kilku elementów, które stanowią duże wyzwanie pod względem poboru prądu. Podana wartość 94.76 mA stanowi średnie zużycie prądu urządzenia będącego w trybie bezczynności, co nie powinno stanowić problemu dla wbudowanego w Nucleo-L476RG problemu. Potwierdza to wykonany pomiar przez tester USB (Rysunek 4.18), wpięty szeregowo pomiędzy prototyp a źródło zasilania, wyświetlający wartość napięcia oraz prądu pobieranego przez urządzenie.

Obraz zawierający tekst, elektronika, Urządzenie elektroniczne, gadżet

Opis wygenerowany automatycznie

Rysunek 4.18 Pomiar prądu pobieranego przez prototyp

Moduł GSM zużywa najwięcej prądu, głównie z powodu możliwości pojawienia się skoków prądowych sięgających do 1.6 A. Zastosowanie superkondesatora o dużej pojemności pozwala na uzupełnienie obsługi skoków prądowych ładunkiem zgromadzonym w superkondensatorze, bez nadmiernego obciążenia regulatora napięcia 5V. Dodatkowo, skoki prądowe występujące podczas obracania zamka przez serwomechanizm są stabilizowane przez stabilizator znajdujący się na płytce developerskiej, który bez problemu obsługuje zwiększony pobór prądu.

## 4.4 Schemat działania

Działanie urządzenia zaprezentowano w formie schematu blokowego na Rysunku 4.19. Dodatkowo, obok działań urządzenia, zaprezentowano komunikaty wyświetlane na wyświetlaczu LCD. Urządzenie pracuje w sposób proceduralny, wykonując kolejną czynność dopiero po zakończeniu poprzedniej, co zapobiega niespodziewanego przechodzenia do innych stanów urządzenia. Podczas inicjalizacji, zachodzącej po włączeniu zasilania urządzenia, użytkownik jest proszony o przesłanie przez Bluetooth numeru telefonu kuriera, na który zostanie wysłana wiadomość SMS z kodem dostępu do skrytki. Następnie, otrzymując kod dostępu, kurier wpisuje go przy próbie dostarczenia lub odebrania przesyłki, aby odblokować zamek i wykonać zamierzoną czynność, po czym potwierdza operację wciskając przycisk ‘A’ na klawiaturze membranowej. Aby odebrać swoją paczkę, użytkownik przesyła komendę ‘open’, co prowadzi do odblokowania skrytki. W celu zapobieżenia dostępu do skrytki przez osoby postronne, użytkownik wysyła komendę ‘close’, która blokuje zamek do czasu kolejnego użycia urządzenia. Gdy użytkownik zechce ponownie skorzystać z urządzenia, przesyła komendę ‘reset’, umożliwiającą powtórzenie całego procesu od początku.

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

Rysunek 4.19 Schemat działania urządzenia oraz wyświetlanych komunikatów

# Rozdział 5 Specyfikacja wewnętrzna – Oprogramowanie

Program został napisany w języku C, wykorzystując środowisko STM32CubeIDE, które opisano w rozdziale 3. Dzięki narzędziu „Device Configuration Tool” oraz bibliotekom HAL, środowisko programistyczne generuje automatycznie obszerny kod potrzebny do obsługi mikrokontrolera. To z kolei umożliwia programiście szybsze rozpoczęcie pracy nad projektowanym systemem. Kod używany w mikrokontrolerze został podzielony na dwa pliki źródłowe: **main.c** i **HD44780.c**. Plik main.c odpowiada za komunikację UART, inicjalizację modułów oraz obsługę klawiatury matrycowej, natomiast HD44780.c jest przeznaczony do zarządzania komunikacją z wyświetlaczem LCD i definiuje strukturę wyświetlanych komunikatów.

## 5.1 Inicjalizacja modułów komunikacyjnych

Inicjalizacja modułów komunikacyjnych mikroprocesora STM32L476RG jest realizowana przy użyciu „Device Configuration Tool”. Dzięki graficznemu interfejsowi użytkownika, możliwy jest wybór pożądanego interfejsu komunikacyjnego oraz dostosowanie wymagających parametrów za pomocą dostępnych opcji (Rysunek 5.1). Dla interfejsów USART1, USART2 oraz USART3 zdecydowano się na wybór trybu asynchronicznego z obsługą przerwań. Prędkość transmisji danych została ustalona na 115200 bit/s dla USART1 i USART2, natomiast dla USART3 – na 9600 bit/s. USART2 pełni funkcję interfejsu debugującego, który transmituje dane do debuggera ST-Link zlokalizowanego na płytce Nucleo, gdzie następuje konwersja sygnału na USB. Dodatkowo zadeklarowano interfejs komunikacyjny I2C, działający w standardowym trybie, z prędkością transmisji danych wynoszącą 100 kHz.

Sygnał PWM generowany jest przez sprzętowy licznik, który wymaga konfiguracji prescalera – dzielącego częstotliwość mikrokontrolera, ustawienia maksymalnej wartości licznika określającej okres powtarzania sygnału, oraz ustalenia wartości odpowiadającej zakresowi sterowania szerokością pulsu, co jest realizowane poprzez wystawienie sygnału na odpowiedni pin wyjściowy [24].

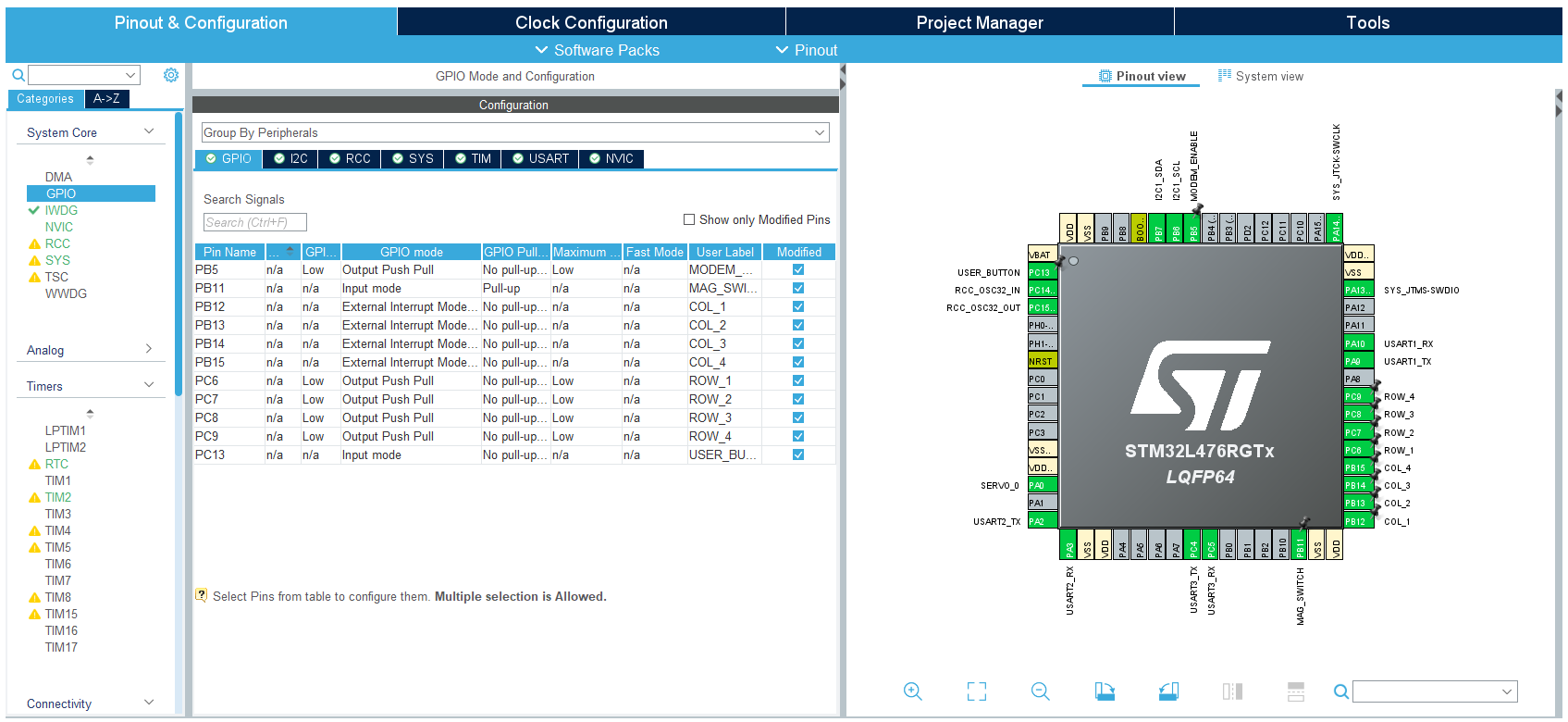
Moduł RNG, który jest wykorzystywany do generowania kodu dostępu, wymaga jedynie aktywacji poprzez zaznaczenie pola 'Activated' oraz włączenia obsługi przerwań. Z kolei konfiguracja systemu IWDG (ang. *Independent Watchdog*) obejmuje ustawienie prescalera i określenie czasu, po upływie którego następuje reset mikrokontrolera.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Rysunek 5.1 Ustawienia interfejsu UART w narzędziu Device Configuration Tool

Najważniejsze ustawienia dostępne w 'Device Configuration Tool' dotyczą deklaracji pinów GPIO. W ramach tego narzędzia możliwe jest określenie trybu pracy pinów jako wejście/wyjście, dodawanie oporników podciągających (Pull-up) lub zwierających (Pull-down), a także przypisywanie własnych oznaczeń poszczególnym pinom. Opcja zarządzania NVIC (ang. *Nested Vectored Interrupt Controller*) umożliwia deklarację obsługi przerwań, jak również pozwala na ustawienie priorytetu dla poszczególnych przerwań [29].



Rysunek 5.2 Ustawienia GPIO w narzędziu Device Configuration Tool

## 5.2 Plik main.c

Komunikacja UART została zrealizowana na zasadzie przerwań. Gdy na wejściu RX mikroprocesora pojawia się nowa informacja, wykrywana przez funkcję HAL\_UART\_Receive\_IT() wykonywana jest funkcja przerwania HAL\_UART\_RxCpltCallback(), która umożliwia identyfikację, na wejściu RX którego UART zaszła zmiana stanów.

|  |  |
| --- | --- |
| 222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239 | **void** **HAL\_UART\_RxCpltCallback**(UART\_HandleTypeDef \*huart)  {  **if**(huart == &huart2)  {  line\_append\_debug(uart2\_rx\_buffer);  HAL\_UART\_Receive\_IT(&huart2, &uart2\_rx\_buffer, 1);  }  **else** **if**(huart == &huart1)  {  line\_append\_gsm(uart1\_rx\_buffer);  HAL\_UART\_Receive\_IT(&huart1, &uart1\_rx\_buffer, 1);  }  **else** **if**(huart == &huart3)  {  line\_append\_bluetooth(uart3\_rx\_buffer);  HAL\_UART\_Receive\_IT(&huart3, &uart3\_rx\_buffer, 1);  }  } |

Rysunek 5.3 Funkcja HAL\_UART\_RxCpltCallback

W zależności od wykrytego UART, wywoływana jest funkcja 'line\_append\_x', która zapisuje otrzymywane znaki do buforów. Po otrzymaniu znaków końca linii <CR><LF>, funkcja przesyła otrzymaną wiadomość na konsolę komputera. W przypadku line\_append\_gsm(), informacja o końcu linii jest wykrywana po wystąpieniu dłuższej przerwy w przesyłaniu danych, jest to spowodowane konstrukcją odpowiedzi zwrotnej modemu GSM która wygląda w następujący sposób: <CR><LF> OK <CR><LF> [30].

Przesyłanie wartości przez UART realizowane jest na dwa sposoby: manualnie, za pośrednictwem konsoli debugującej, oraz automatycznie. Automatycznie jest wysyłana wiadomość SMS do modułu GSM ze względu na konieczność zachowania kolejności wysyłanych komend oraz znaku 0x1A oznaczającego koniec wiadomości. Przesyłanie wiadomości przez wyjście TX za pomocą funkcji HAL\_UART\_Transmit\_IT() inicjuje wywołanie funkcji przerwania HAL\_UART\_TxCpltCallback(). W tej funkcji, w zależności od wykrytego UART, możliwe jest wykonanie dodatkowych operacji. W przypadku UART2, do przesyłanej wiadomości dodawany jest dodatkowo znak końca linii <CR><LF>, co ułatwia odczytanie wiadomości na konsoli komputera. Jeżeli zostanie wysłany numer telefonu, na który ma być wysłana wiadomość SMS, zapala się flaga if\_phone\_number\_set\_latch która odblokowuje funkcję send\_SMS() (Rysunek 5.5 w Źródłach);

|  |  |
| --- | --- |
| 315  316  317  318  319  320  321  322  323  324  325  326  327  328 | **void** **HAL\_UART\_TxCpltCallback**(UART\_HandleTypeDef \*huart)  {  **if**(huart == &huart1)  {  **if**(if\_phone\_number\_set\_latch == true)  {  send\_SMS();  }  }  **if**(huart == &huart2)  {  send\_end\_line();  }  } |

Rysunek 5.4 Funkcja HAL\_UART\_TxCpltCallback

W funkcji send\_SMS() zastosowano opóźnienie przy użyciu funkcji delay(), która realizuje operację \_\_NOP(). Wybór tego rozwiązania podyktowany był faktem, że dłuższe opóźnienia realizowane przez funkcję HAL\_Delay() mogą blokować wykonywanie innych przerwań co prowadzi do zablokowania mikrokontrolera. Jest to spowodowane wykorzystaniem przez HAL\_Delay() funkcji HAL\_GetTick(), która, pobiera aktualny czas działania mikrokontrolera, korzystając z przerwań blokujących procesor.

|  |  |
| --- | --- |
| 456  457  458  459  460  461  462 | **void** **delay**(uint32\_t iterations)  {  **while**(iterations-- > 0)  {  \_\_NOP();  }  } |

Rysunek 5.6 Funkcja delay()

Ostatnim elementem kodu, który wykorzystuje przerwania, jest obsługa klawiatury matrycowej. Piny wierszy są wyjściami, a piny podłączone do kolumn klawiatury są ustawione w trybie 'External Interrupt Mode', z aktywacją przerwania na zboczu narastającym. Po naciśnięciu przycisku na klawiaturze wywoływana jest funkcja przerwania HAL\_GPIO\_EXTI\_Callback(), która zawiera implementację mechanizmu debouncing, zapobiegającego przypadkowym, wielokrotnym wciśnięciom przycisku. Następnie, zmienia się konfiguracja pinów kolumn, które są ustawiane w tryb wejściowy (input), ale bez aktywacji przerwań. W celu wykrycia naciśniętego przycisku, realizowane jest przemiatanie pinów wierszy: w określonym momencie stan wysoki jest przydzielany tylko jednemu wierszowi. Jeśli w wyniku tej operacji zostanie wykryte zwarcie na konkretnym przycisku, sygnał ten jest identyfikowany na pinie wejściowym odpowiedniej kolumny. Po skończonym przemiataniu zostaje ustawiony stan wysoki na każdym pinie wiersza a piny kolumn wracają do trybu wykrywania przerwań, a wartość wciśniętego przycisku jest zapisywana do bufora.

|  |  |
| --- | --- |
| 346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362 | HAL\_GPIO\_WritePin(ROW\_1\_GPIO\_Port, ROW\_1\_Pin, 1);  HAL\_GPIO\_WritePin(ROW\_2\_GPIO\_Port, ROW\_2\_Pin, 0);  HAL\_GPIO\_WritePin(ROW\_3\_GPIO\_Port, ROW\_3\_Pin, 0);  HAL\_GPIO\_WritePin(ROW\_4\_GPIO\_Port, ROW\_4\_Pin, 0);  **if**(GPIO\_Pin == COL\_1\_Pin && HAL\_GPIO\_ReadPin(COL\_1\_GPIO\_Port, COL\_1\_Pin))  {  pressed\_key = '1';  } **else** **if**(GPIO\_Pin == COL\_2\_Pin && HAL\_GPIO\_ReadPin(COL\_2\_GPIO\_Port, COL\_2\_Pin))  {  pressed\_key = '2';  } **else** **if**(GPIO\_Pin == COL\_3\_Pin && HAL\_GPIO\_ReadPin(COL\_3\_GPIO\_Port, COL\_3\_Pin))  {  pressed\_key = '3';  } **else** **if**(GPIO\_Pin == COL\_4\_Pin && HAL\_GPIO\_ReadPin(COL\_4\_GPIO\_Port, COL\_4\_Pin))  {  pressed\_key = 'A';  } |

Rysunek 5.7 Fragment przedstawiający przemiatanie wierszy

Funkcja main() rozpoczyna się od początkowych inicjalizacji modułów, które obejmują: włączenie pinu Enable modemu, inicjalizację wyświetlacza LCD za pomocą funkcji lcd\_init() z pliku HD44780.c, ustawienie podświetlenia wyświetlacza przy użyciu lcd\_backlight() oraz czyszczenie wyświetlacza z ewentualnej niepożądanej zawartości za pomocą lcd\_clear(). Dodatkowo inicjalizowany jest timer PWM, który steruje serwomechanizmem, z wymuszeniem ustawienia na pozycję otwartego zamka. Dodatkowo, bufor zapisujący wpisywany klucz dostępu jest czyszczony, na pinach wierszy klawiatury matrycowej ustawiany jest stan wysoki, a także rozpoczyna się wykrywanie otrzymywanych znaków na pinach RX wejść UART. Druga część funkcji main() obejmuje pętlę while(1), w której cyklicznie wykonywane są funkcje: check\_timeout\_gsm(), przeznaczona do wykrywania zakończenia transmisji wiadomości przez moduł GSM, lcd\_display(), rozbudowana funkcja wyświetlająca komunikaty w zależności od aktualnego stanu urządzenia, oraz lcd\_display\_key(), odpowiedzialna za prezentację wpisywanego kodu dostępu. Dodatkowo, odbywa się reset Watchdoga za pomocą funkcji HAL\_IWDG\_Refresh().

|  |  |
| --- | --- |
| 503  504  505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524 | HAL\_GPIO\_WritePin(MODEM\_ENABLE\_GPIO\_Port, MODEM\_ENABLE\_Pin, *GPIO\_PIN\_SET*);  HAL\_Delay(1000);  lcd\_init();  send\_AT\_init();  lcd\_backlight(1);  lcd\_clear();  HAL\_TIM\_PWM\_Start(&htim2, TIM\_CHANNEL\_1);  \_\_HAL\_TIM\_SET\_COMPARE(&htim2, TIM\_CHANNEL\_1, 1600);  **memset**(key\_buffer, '\0', BUFFER\_SIZE);  HAL\_GPIO\_WritePin(ROW\_1\_GPIO\_Port, ROW\_1\_Pin, 1);  HAL\_GPIO\_WritePin(ROW\_2\_GPIO\_Port, ROW\_2\_Pin, 1);  HAL\_GPIO\_WritePin(ROW\_3\_GPIO\_Port, ROW\_3\_Pin, 1);  HAL\_GPIO\_WritePin(ROW\_4\_GPIO\_Port, ROW\_4\_Pin, 1);  HAL\_UART\_Receive\_IT(&huart2, &uart2\_rx\_buffer, 1);  HAL\_UART\_Receive\_IT(&huart1, &uart1\_rx\_buffer, 1);  HAL\_UART\_Receive\_IT(&huart3, &uart3\_rx\_buffer, 1); |

Rysunek 5.8 Część inicjalizacyjna funkcji main()

|  |  |
| --- | --- |
| 530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546  547 | **while** (1)  {  check\_timeout\_gsm();  lcd\_display(if\_phone\_number\_set\_latch, if\_key\_pressed, key\_buffer, access\_key, pressed\_key, open\_close\_cmd);  if\_key\_pressed = lcd\_display\_key(key\_buffer, if\_key\_pressed);  **if**(pressed\_key == 'D')  {  reset\_buffer();  }  HAL\_IWDG\_Refresh(&hiwdg);  /\* USER CODE END WHILE \*/  /\* USER CODE BEGIN 3 \*/  } |

Rysunek 5.9 Pętla while funkcji main()

## 5.3 Plik HD44780.c

W przeciwieństwie do pliku main.c, plik HD44780.c nie zawiera kodu wygenerowanego przez „Device Configuration Tool”, gdyż został w całości opracowany przez użytkownika. W pliku HD44780.c zawarte są funkcje odpowiedzialne za obsługę komunikacji z wyświetlaczem LCD i zarządzanie poszczególnymi stanami urządzenia.

Kluczową funkcją w pliku HD44780.c jest lcd\_write\_nibble(), która służy do przesyłania danych do wyświetlacza LCD poprzez interfejs I2C. W związku z wykorzystaniem trybu 4-bitowego w komunikacji z wyświetlaczem, konieczne jest przesyłanie danych oraz komend w formie dwóch półbajtów (ang. nibble). Zostało to wyjaśnione w rozdziale 4.

|  |  |
| --- | --- |
| 37  38  39  40  41  42  43  44  45  46  47 | **static** **void** **lcd\_write\_nibble**(uint8\_t nibble, uint8\_t rs)  {  uint8\_t data = nibble << D4\_BIT;  data |= rs << RS\_BIT;  data |= backlight\_state << BL\_BIT;  data |= 1 << EN\_BIT;  HAL\_I2C\_Master\_Transmit(&hi2c1, I2C\_ADDR << 1, &data, 1, TIMEOUT);  HAL\_Delay(1);  data &= ~(1 << EN\_BIT);  HAL\_I2C\_Master\_Transmit(&hi2c1, I2C\_ADDR << 1, &data, 1, TIMEOUT);  } |

Rysunek 5.10 Funkcja lcd\_write\_nibble()

Funkcja lcd\_write\_nibble() akceptuje dwa argumenty: nibble, co oznacza półbajt przeznaczony do transmisji, oraz rs, który jest bitem wyboru rejestru. Bit ten określa, czy przekazywany półbajt stanowi komendę, czy dane do wyświetlenia. Ramka przeznaczona do przesłania informacji jest budowana poprzez najpierw pozycjonowanie wysyłanego półbajtu w odpowiednim miejscu. Następnie ustawiane są: bit rs (wyboru rejestru), bit podświetlenia, oraz bit Enable. Po skompletowaniu ramki z odpowiednio ustawionymi bitami, jest ona przesyłana przy użyciu funkcji HAL\_I2C\_Master\_Transmit(). Następuje zadeklarowanie krótkiego opóźnienia HAL\_Delay(), które pozwala układowi HD44780 na przetworzenie otrzymanych danych. Na zakończenie, przesyłany jest wyłączony bit Enable, zapobiegając ewentualnemu przetworzeniu niepożądanych stanów na linii danych.

Inną ważną, a zarazem prostą funkcją jest lcd\_send\_cmd(), która odpowiada za formowanie i wysyłanie komendy podzielonej na dwa półbajty. Dla komend 0x01 (czyszczenie wyświetlacza) i 0x02 (powrót kursora do domyślnej pozycji) dodane jest specjalne opóźnienie, umożliwiające kontrolerowi wyświetlacza bezproblemowe przetworzenie tych komend. Funkcja lcd\_send\_data(), konstruowana jest w podobny sposób, jednak nie zawiera dodatkowych opóźnień dla wyjątkowych komend.

|  |  |
| --- | --- |
| 49  50  51  52  53  54  55  56  57  58  59 | **void** **lcd\_send\_cmd**(uint8\_t cmd)  {  uint8\_t upper\_nibble = cmd >> 4;  uint8\_t lower\_nibble = cmd & 0x0F;  lcd\_write\_nibble(upper\_nibble, 0);  lcd\_write\_nibble(lower\_nibble, 0);  **if**(cmd == 0x01 || cmd == 0x02)  {  HAL\_Delay(2);  }  } |

Rysunek 5.11 Funkcja lcd\_send\_cmd()

Pozostałe funkcje, takie jak lcd\_init(), lcd\_write\_string(), lcd\_set\_cursor() oraz lcd\_clear(), są funkcjami prostymi, które korzystają z komend przeznaczonych dla kontrolera HD44780. Komendy te są przesyłane za pomocą wcześniej omówionych funkcji.

Najbardziej rozbudowaną funkcją, która implementuje znaczącą część programu, jest lcd\_display(). Funkcja ta przyjmuje sześć argumentów: current\_state\_telephone, aby określić, czy numer telefonu został już przesłany; current\_state\_key, wskazujący, czy klucz dostępu został wprowadzony; key\_buffer, zawierający wpisany kod do weryfikacji z wygenerowanym kluczem dostępu access\_key; pressed\_button, do wykrycia, czy został wciśnięty przycisk ‘A’; oraz ble\_cmd, który służy do przekazywania komend otrzymywanych przez Bluetooth.

Kod zawarty w funkcji lcd\_display() jest zorganizowany w cztery główne sekcje:

* Deklaracje zmiennych wewnętrznych oraz komunikatów do wyświetlania
* Weryfikacja zamknięcia drzwi w celu zapobiegania zamknięciu zamka, gdy drzwi są otwarte oraz wyświetlanie odpowiedniego komunikatu
* Sprawdzenie aktualnego stanu urządzenia, aby wyświetlić właściwy komunikat
* Mechanizm przełączania stanów dla wyświetlanych komunikatów LCD.

|  |  |
| --- | --- |
| 154  155  156  157  158  159  160  161  162  163  164  165  166 | **static** **char** \*door\_open = "Zamek otwarty";  **static** **char** \*door\_closed = "Zamek zamkniety";  **static** **char** \*telephone\_not\_set = "Wprowadz nr. tel.";  **static** **char** \*telephone\_set = "Wpisz kod: ";  **static** **char** \*key\_good = "Klucz poprawny";  **static** **char** \*key\_bad = "Zly klucz";  **static** **char** \*message\_1 = "Nacisnij A po ";  **static** **char** \*message\_2 = "odebraniu paczki";  **static** **char** \*unlock = "Odblokuj";  **static** **char** \*lock = "Zablokuj";  **static** **char** \*reset = "Zresetuj";  **static** **char** \*close\_door = "Zamknij drzwi";  **static** **char** \*clear\_line = " "; |

Rysunek 5.12 Deklaracje komunikatów lcd\_display()

|  |  |
| --- | --- |
| 174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193 | **if**(HAL\_GPIO\_ReadPin(MAG\_SWITCH\_GPIO\_Port, MAG\_SWITCH\_Pin) == *GPIO\_PIN\_RESET*)  {  **switch**(back\_condition)  {  **case** *BACK\_2*:  lcd\_number = *LCD\_2*;  back\_condition = *NO\_BACK*;  **break**;  **case** *BACK\_6*:  lcd\_number = *LCD\_6*;  back\_condition = *NO\_BACK*;  **break**;  **case** *BACK\_8*:  lcd\_number = *LCD\_8*;  back\_condition = *NO\_BACK*;  **break**;  **default**:  **break**;  }  } |

Rysunek 5.13 Część kodu weryfikującego zamknięcie drzwi

|  |  |
| --- | --- |
| 276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303 | **case** *LCD\_2*:  **if**(HAL\_GPIO\_ReadPin(MAG\_SWITCH\_GPIO\_Port, MAG\_SWITCH\_Pin) == *GPIO\_PIN\_SET*)  {  lcd\_clear();  lcd\_set\_cursor(0, 0);  lcd\_write\_string(close\_door);  back\_condition = *BACK\_2*;  } **else**  {  lcd\_set\_cursor(0, 0);  lcd\_write\_string(clear\_line);  lcd\_set\_cursor(0, 0);  lcd\_write\_string(door\_closed);  lcd\_set\_cursor(1, 0);  lcd\_write\_string(clear\_line);  lcd\_set\_cursor(1, 0);  lcd\_write\_string(telephone\_set);  delay(delay\_second\*2);  \_\_HAL\_TIM\_SET\_COMPARE(&htim2, TIM\_CHANNEL\_1, 650);  can\_enter\_key = true;  }  lcd\_number = *DONE*;  **break**; |

Rysunek 5.14 Fragment mechanizmu przełączającego wyświetlany komunikat

# Rozdział 6 Weryfikacja i walidacja

* sposób testowania w ramach pracy (np. odniesienie do modelu V)
* organizacja eksperymentów
* przypadki testowe, zakres testowania (pełny/niepełny)
* wykryte i usunięte błędy
* opcjonalnie wyniki badań eksperymentalnych

|  |  |
| --- | --- |
| Tabela 4.2. Nagłówek tabeli jest nad tabelą. | |
| Poziom 1 | 24 pt |
| Poziom 2 | 20 pt |
| Poziom 3 | 16 pt |
|  | |

# Rozdział 7 Podsumowanie i wnioski

* uzyskane wyniki w świecie postawionych celów i zdefiniowanych wyżej wymagań
* kierunki ewentualnych danych prac (rozbudowa funkcjonalna …)
* problemy napotkane w trakcie pracy

# Bibliografia

1. Drew McKay. <https://houstonlawreview.org/article/30086-the-porch-pirate-problem> (dostęp 24.01.2024)
2. <https://www.vivint.com/resources/article/porch-pirates> (dostęp 26.01.2024)
3. <https://www.security.org/package-theft/annual-report/> (dostęp 27.01.2024)
4. <https://inpost.pl/o-inpost> (dostęp 27.01.2024)
5. <https://lockermap.com> (dostęp 27.01.2024)
6. By Sikander Iqbal, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=81749361> (dostęp 30.01.2024)
7. <https://os.mbed.com/platforms/ST-Nucleo-L476RG/> (dostęp 30.01.2024)
8. <https://www.st.com/content/st_com/en/stm32cubeide.html> (dostęp 30.01.2024)
9. <https://www.nordicsemi.com/Products/Development-tools/nrf-connect-for-mobile> (dostęp 30.01.2024)
10. <https://botland.com.pl/stm32-nucleo/18800-stm32-nucleo-l476rg-z-mcu-stm32l476rgt6-arm-cortex-m4-5904422364977.html> (dostęp 31.01.2024)
11. <https://forbot.pl/blog/kurs-stm32l4-platforma-sprzetowa-i-srodowisko-stm32cubeide-id46575> (dostęp 31.01.2024)
12. Nota katalogowa STM32L476RGT6 <https://www.st.com/en/microcontrollers-microprocessors/stm32l476rg.html> (dostęp 31.01.2024)
13. <https://botland.com.pl/moduly-gsm/13924-modul-2ggsm-u-gsm-shield-v219-m95fa-do-arduino-i-raspberry-pi-zlacze-ufl-5904422321727.html> (dostęp 31.01.2024)
14. <https://sklep.msalamon.pl/produkt/antena-do-modulu-gsm-sim800l/> (dostęp 31.01.2024)
15. <https://html.alldatasheet.com/html-pdf/1179058/ETC1/HM-10/109/1/HM-10.html> (dostęp 31.01.2024)
16. <https://kamami.pl/45967-large_default/at-09hm-10-modul-bluetooth-40-ble.jpg> (dostęp 31.01.2024)
17. <https://botland.com.pl/konwertery-pozostale/2352-konwerter-i2c-dla-wyswietlacza-lcd-hd44780-5903351248693.html> (dostęp 31.01.2024)
18. Nota katalogowa PCF8574 <https://pdf1.alldatasheet.com/datasheet-pdf/view/525245/NXP/PCF8574T.html> (dostęp 31.01.2024)
19. Nota katalogowa HD44780 <https://pdf1.alldatasheet.com/datasheet-pdf/view/63673/HITACHI/HD44780.html> (dostęp 31.01.2024)
20. <https://botland.com.pl/wyswietlacze-alfanumeryczne-i-graficzne/224-wyswietlacz-lcd-2x16-znakow-niebieski-5903351241380.html> (dostęp 31.01.2024)
21. <https://botland.com.pl/klawiatury-arduino/17119-klawiatura-membranowa-samoprzylepna-4x4-16-klawiszy-5903351247870.html> (dostęp 31.01.2024)
22. <https://forbot.pl/blog/leksykon/serwomechanizm> (dostęp 31.01.2024)
23. <https://botland.com.pl/serwomechanizmy/484-serwo-towerpro-sg-90-micro-180-stopni-5904422329488.html> (dostęp 1.02.2024)
24. <https://msalamon.pl/jak-pomachac-swoim-orczykiem-stm32-spotyka-sie-z-serwem/> (dostęp 1.02.2024)
25. <https://pl.wikipedia.org/wiki/Kontaktron> (dostęp 1.02.2024)
26. Autorstwa André Karwath aka Aka - Praca własna, CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=124713> (dostęp 1.02.2024)
27. <https://www.st.com/content/ccc/resource/technical/layouts_and_diagrams/schematic_pack/group2/5a/85/d6/9a/34/e2/47/1d/MB1136-DEFAULT-C05_Schematic/files/MB1136-DEFAULT-C05_Schematic.pdf/jcr:content/translations/en.MB1136-DEFAULT-C05_Schematic.pdf> (dostęp 1.02.2024)
28. <https://www.tme.eu/Document/28dbf8e74f9375a8188991ee77269e04/LD1117.pdf> (dostęp 1.02.2024)
29. <https://www.st.com/resource/en/product_training/STM32WB-System-Nested-Vectored-Interrupt-Control-NVIC.pdf> (dostęp 01.02.2024)
30. <https://itbrainpower.net/downloadables/Quectel_M95_AT_Commands_Manual_V3.2.pdf> (dostęp 02.02.2024)

Dodatki

# Spis skrótów i symboli

SMS Krótka wiadomość tekstowa (ang. *Short Message Service*)

LCD Wyświetlacz ciekłokrystaliczny (ang. *Liquid-Crystal Display*)

GSM (ang. *Global System for Mobile Communications*)

HAL (ang. *Hardware Abstraction Layer*)

UART (ang. *Universal Asynchronous Receiver-Transmitter*)

USB (ang. *Universal Serial Bus*)

I2C (ang. *Inter-Integrated Circuit*)

PWM (ang. *Pulse-Width Modulation*)

RNG (ang. *Random Number Generator*)

GPIO (ang. *General-Purpose Input/Output*)

GND Masa (ang. *ground*)

SMA (ang. *SubMiniature version A*)

SIM (ang. *Subscriber Identity Module*)

Li-Po (*litowo-polimerowy*)

Li-Ion (*litowo- jonowy*)

RX (ang. *receiver*)

TX (ang. *transmitter*)

SDA (ang. *Serial Data Line*)

SCL (ang. *Serial Clock Line*)

SIPO (ang. *Serial-In Parallel-Out*)

LDO (ang. *Low-Dropout Regulator*)

IWDG (ang. *Independent Watchdog*)

NVIC (ang. *Nested Vectored Interrupt Controller*)

# Źródła

|  |  |
| --- | --- |
| 266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304 | **void** **send\_SMS**(**void**)  {  **static** **char** message\_cmgf[] = "AT+CMGF=1\r";  **static** **char** message\_cscs[] = "AT+CSCS=\"GSM\"\r";  **static** **char** message\_cmgs[31];  **sprintf**(message\_cmgs, "AT+CMGS=\"+48%s\"\r", phone\_number);  access\_key\_draw();  **static** **char** message\_message[34];  **sprintf**(message\_message, "Kod dostepu do skrytki: %s", access\_key);  **static** **char** message\_ctrlz = 0x1A;  delay(100);  **switch**(message\_number)  {  **case** *MESSAGE\_1*:  HAL\_UART\_Transmit\_IT(&huart1, (uint8\_t\*)message\_cmgf, **strlen**(message\_cmgf));  message\_number = *MESSAGE\_2*;  **break**;  **case** *MESSAGE\_2*:  HAL\_UART\_Transmit\_IT(&huart1, (uint8\_t\*)message\_cscs, **strlen**(message\_cscs));  message\_number = *MESSAGE\_3*;  **break**;  **case** *MESSAGE\_3*:  HAL\_UART\_Transmit\_IT(&huart1, (uint8\_t\*)message\_cmgs, **strlen**(message\_cmgs));  message\_number = *MESSAGE\_4*;  **break**;  **case** *MESSAGE\_4*:  HAL\_UART\_Transmit\_IT(&huart1, (uint8\_t\*)message\_message, **strlen**(message\_message));  message\_number = *MESSAGE\_5*;  **break**;  **case** *MESSAGE\_5*:  HAL\_UART\_Transmit\_IT(&huart1, (uint8\_t\*)&message\_ctrlz, 1);  message\_number = *DONE*;  **break**;  **default**:  **break**;  }  } |

Rysunek 5.5 Funkcja send\_SMS()

# Lista dodatkowych plików, uzupełniających tekst pracy

W systemie, do pracy dołączono dodatkowe pliki zawierające:

* źródła programu,
* dane testowe
* film pokazujący działanie opracowanego oprogramowania lub zaprojektowanego i wykonanego urządzenia,
* itp.

# Spis rysunków

2.1 „Paczkomat®” firmy InPost 5

2.2 Automat paczkowy „Amazon Hub Locker” firmy Amazon [6] 6

2.3 „Paczkoport S-1N SZ” firmy Paczkoport 7

3.1 Perspektywa „Device Configuration Tool” programu STM32CubeIDE 10

3.2 Perspektywa „C/C++” programu STM32CubeIDE 10

3.3 Perspektywa „Debug” programu STM32CubeIDE 11

3.4 Program Tera Term 11

3.5 Aplikacja nRF Connect 12

4.1 Schemat podłączenia modułów prototypu 13

4.2 Nucleo-L476RG [10] 15

4.3 Opis wyprowadzeń GPIO płytki Nucleo-L476RG [11] 16

4.4 Moduł 2G „u-GSM shield” [13] 17

4.5 Antena GSM z przewodem SMA – u.FL [14] 17

4.6 Moduł Bluetooth HM-10 [16] 18

4.7 Konwerter I2C [17] 18

4.8 Schemat blokowy układu PCF8574 [18] 19

4.9 Transmisja I2C konwertera PCF8574 [18] 19

4.10 Mapa adresowa układu PCF8574 [18] 19

4.11 Wyświetlacz alfa-numeryczny LCD 2x16 [20] 20

4.12 Transmisja układu HD44780 w trybie 4-bitowym 20

4.13 Klawiatura membranowa 4x4 [21] 21

4.14 Schemat klawiatury matrycowej 21

4.15 Serwomechanizm TowerPro SG-90 [23] 22

4.16 Przykładowy przebieg PWM sterującego serwomechanizmem [24] 22

4.17 Kontaktron [26] 23

4.18 Pomiar prądu pobieranego przez prototyp 24

4.19 Schemat działania urządzenia oraz wyświetlanych komunikatów 25

5.1 Ustawienia interfejsu UART w narzędziu Device Configuration Tool 27

5.2 Ustawienia GPIO w narzędziu Device Configuration Tool 27

5.3 Funkcja HAL\_UART\_RxCpltCallback 28

5.4 Funkcja HAL\_UART\_TxCpltCallback 29

5.5 Funkcja send\_SMS() Źródła

5.6 Funkcja delay() 29

5.7 Fragment przedstawiający przemiatanie wierszy 30

5.8 Część inicjalizacyjna funkcji main() 30

5.9 Pętla while funkcji main() 31

5.10 Funkcja lcd\_write\_nibble() 31

5.11 Funkcja lcd\_send\_cmd() 32

5.12 Deklaracje komunikatów lcd\_display() 33

5.13 Część kodu weryfikującego zamknięcie drzwi 33

5.14 Fragment mechanizmu przełączającego wyświetlany komunikat 34

# Spis tablic

4.1 Lista użytych komponentów do stworzenia prototypu 16

4.2 Zapotrzebowanie prądowe prototypu 23