POLITECHNIKA GDAŃSKA
WYDZIAŁ ELEKTRONIKI, TELEKOMUNIKACJI I INFORMATYKI

Imię i nazwisko studenta: Adrianna Piekarska
Nr albumu: 165152
Studia pierwszego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Informatyka
Profil: Architektura systemów komputerowych

Imię i nazwisko studenta: Grzegorz Wąs
Nr albumu: 165464
Studia pierwszego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Informatyka
Profil: Inteligentne systemy interaktywne

PROJEKT DYPLOMOWY INŻYNIERSKI

Tytuł projektu w języku polskim: Bezprzewodowy system dostępu do pomieszczeń

Tytuł projektu w języku angielskim: Wireless access control system

Potwierdzenie przyjęcia projektu	
Opiekun projektu *podpis*	Kierownik Katedry/Zakładu (pozostawić właściwe) *podpis*
dr inż. Tomasz Dziubich	

Data oddania projektu do dziekanatu:

Streszczenie

Postępująca miniaturyzacja systemów wbudowanych wynikająca z wykładniczego rozwoju technologii półprzewodnikowych oraz rosnąca ich dostępność sprawiają, że systemy informatyczne znajdują coraz szersze zastosowanie w wielu dziedzinach. Nowoczesne i wydajne systemy stopniowo zastępują tradycyjne rozwiązania sprzed ery informatyzacji. Nowe możliwości niosą ze sobą jednak równie wiele wyzwań, szczególnie z zakresu bezpieczeństwa. Jedną z dziedzin o szerokich perspektywach rozwoju jest bezpieczeństwo przestrzeni wraz z ich uzytkownikami. Niniejsza praca opisuje projekt i implementację bezprzewodowego systemu dostępu do pomieszczeń. Omawia architekturę rozwiązania z uwzględnieniem poszczególnych podsystemów, przedstawia ciekawe aspekty realizacji projektu oraz jego rezultaty. Ponadto, prezentuje zagadnienia związane z bezpieczeństwem oraz wydajnością energetyczną bezprzewodowych systemów opartych na mikrokontrolerach.

TBD - rozszerzyć

Słowa kluczowe: zamek elektroniczny, mikrokontroler, kontrola dostępu, WiFi, RFID, sieć bezprzewodowa, autoryzacja

Dziedzina nauki i techniki, zgodnie z wymogami OECD:

Abstract

Postępująca miniaturyzacja systemów wbudowanych wynikająca z wykładniczego rozwoju technologii półprzewodnikowych oraz rosnąca ich dostępność sprawiają, że systemy informatyczne znajdują coraz szersze zastosowanie w wielu dziedzinach. Nowoczesne i wydajne systemy stopniowo zastępują tradycyjne rozwiązania sprzed ery informatyzacji. Nowe możliwości niosą ze sobą jednak równie wiele wyzwań, szczególnie z zakresu bezpieczeństwa. Jedną z dziedzin o szerokich perspektywach rozwoju jest bezpieczeństwo przestrzeni wraz z ich uzytkownikami. Niniejsza praca opisuje projekt i implementację bezprzewodowego systemu dostępu do pomieszczeń. Omawia architekturę rozwiązania z uwzględnieniem poszczególnych podsystemów, przedstawia ciekawe aspekty realizacji projektu oraz jego rezultaty. Ponadto, prezentuje zagadnienia związane z bezpieczeństwem oraz wydajnością energetyczną bezprzewodowych systemów opartych na mikrokontrolerach.

Keywords:

Spis treści

Spis rysunków

Spis tablic

Wykaz ważniejszych oznaczeń i skrótów

Pojęcie	Wyjaśnienie
Punkt dostępu	Fizyczne zabezpieczenie chroniące przed nieuprawnionym dostępem, przykładowo: zamek, bramka
RFID	Technologia wykorzystująca fale radiowe w celu przesyłania danych (ang. Radio-frequency identification)
Identyfikator	Identyfikator RFID. Inne określenia: karta, token, tag
Kontroler	Komponent odpowiedzialny za zarządzanie układem zamka
Break glass device	Urządzenie awaryjne umożliwiające odcięcie zasilania w zamku

Rozdział 1

Wstep i cel pracy

1.1 Wstęp

Zapewnienie bezpieczeństwa przestrzeni użytkowych i osób z nich korzystających stanowi kluczowy aspekt zarządzania obiektami zarówno publicznymi, jak i prywatnymi. Dzięki zastosowaniu odpowiedniej infrstruktury, bezpieczeństwo osób przebywających na terenie obiektu rośnie, a ryzyko kradzieży lub zniszczenia mienia przez niepowołane osoby spada. Podstawową metodą kontroli dostępu do pomieszczeń jest montaż urządzeń ryglujących z układem zapadkowym rozpoznającym fizyczny klucz. Jednak ze względu na postępujący rozwój technologiczny, w obecnych czasach coraz częściej stosowane są systemy oparte na uwierzytelnianiu elektronicznym.

Pod względem celu i ogólnych zasad działania elektroniczny system kontroli dostępu do pomieszczeń nie rózni się od swojego tradycyjnego odpowiednika. Głównym celem jest autoryzacja prób dostępu użytkowników na podstawie kluczy w taki sposób, aby dostęp został przyznany tylko osobie posiadającej powiązany z danym punktem dostępu klucz.

Przewagę systemów opartych na urządzeniach elektronicznych nad systemami czysto mechanicznymi stanowią cechy takie jak łatwość obsługi czy możliwość zdalnego zarządzania oraz zbierania danych i monitorowania prób dostępu w celu późniejszej analizy.

1.2 Cel i zakres pracy

Celem niniejszej pracy jest projekt oraz implementacja sytemu dostępu do pomieszczeń z wykorzystaniem technologii takich jak Wi-Fi oraz RFID (ang. *Radio-frequency identification*), w którym podmiotem odpowiedzialnym za autoryzację prób dostępu jest serwer, a komunikacja pomiędzy układem sterującym zamkiem a podsystemem autoryzacji jest realizowana bezprzewodowo. Może on znaleźć zastosowanie jako łatwy w instalacji i obsłudze, lekki i wydajny system dla małych i średnich obiektów.

Założenia dotyczące opisywanego systemu są następujące:

1. Logika uwierzytelniania powinna być zaimplementowana na serwerze. Urządzenie klienckie

(zamek) powinno pełnić jedynie rolę pośrednika w tym procesie.

2. Komunikacja pomiędzy urządzeniem klienckim (układem zamka) a serwerem powinna odbywać się bezprzewodowo.
3. System powinien być wydajny energetycznie i umożliwiać operację zamków na zasilaniu bateryjnym.
4. System powinien implementować niezbędne mechanizmy bezpieczeństwa.

Pracę nad systemem prowadziły dwie osoby. W ramach tej pracy powstały:

- Protoyp układu zamka,
- Oprogramowanie serwera uwierzytelniania,
- Aplikacja do zarządzania,
- Baza danych.

Implementacja prototypu obejmowała stworzenie pojedynczego układu zamka. Ze względu na prototypowy charakter pracy nie przetestowano działania systemu z większą liczbą zamków. Nie ma jednak powodów by twierdzić, że po minimalnych modyfikacjach system nie działałby poprawnie z większą liczbą zamków.

Tutaj podział pracy i obowiązków - TBD

1.3 Struktura pracy

Rozdział 1 stanowi wstęp do pracy, określa jej cel i strukturę. Rozdział 2 przedstawia dziedzinę problemu, przywołuje najważniejsze definicje związane z tematem, dokonuje przedstawienia i porównania kilku istniejących rozwiązań, a także opisu podstawowych zagrożeń bezpieczeństwa, z którymi spotkali się autorzy podczas pracy nad systemem. Rozdział 3 prezentuje projekt rozwiązania. Rozdziały 4 oraz 5 przedstawiają wynik oraz przebieg procesu implementacyjnego wraz z najciekawszymi problemami oraz przeglądem wykorzystanych technologii odpowiednio po stronie układu zamka oraz serwera. Pracę zamyka rozdział 6, który opisuje rezultaty pracy nad projektem, przedstawia możliwe rozszerzenia prototypu oraz dokonuje podsumowania, włącznie z oceną użyteczności, bezpieczeństwa oraz spełnienia wymagań.

Rozdział 2

Dziedzina problemu

Niniejszy rozdział krótko opisuje dziedzinę problemu w oderwaniu od szczegółów technicznych przygotowanego w ramach pracy rozwiązania, nakreśla podstawowe zagadnienia związane z bezpieczeństwem tego typu systemów, a także przedstawia porównanie niektórych z obecnie dostępnych na rynku systemów kontroli dostępu.

2.1 Kontrola dostępu

Kontrola dostępu definiowana jest w [1] jako środki mające na celu zapewnienie, że do zasobów systemu przetwarzania danych mogą mieć dostęp tylko uprawnione jednostki w uprawniony sposób.

British Security Industry Association wyodrębnia kilka komponentów składających się na system kontroli dostępu [2]. Poniżej przedstawiono najważniejsze z nich.

Poświadczenie tożsamości (ang. *credentials*) to fizyczny lub materialny obiekt, element wiedzy lub cecha biometryczna umożliwiająca uzyskanie dostępu do kontrolowanej strefy. Najczęściej jako poświadczenie tożsamości stosuje się kody, np. PIN (ang. *Personal Identification Number*, osobisty numer identyfikacyjny), tokeny (karty, urządzenia mobilne itp.) oraz dane biometryczne.

British Security Industry Association terminem ”czytniki” (ang. *readers*) nazywa urządzenia odpowiedzialne za kontrolę dostępu. Dla uproszczenia nomenklatura ta została zachowana w niniejszej sekcji. W innych częściach niniejszej pracy termin ”czytnik” używany jest w znaczeniu urządzenia odpowiedzialnego wyłącznie za odczyt danych z nośnika.

W większości przypadków tylko wejście podlega kontroli. Aby możliwa była również kontrola wyjścia z chronionego terenu, potrzebny jest drugi czytnik umieszczony po drugiej stronie drzwi. Jeżeli obustronna kontrola nie jest wymagana, stosuje się zazwyczaj przycisk umożliwiający otworzenie zamka od środka.

Urządzenia wyjściowe (ang. *egress devices*) umożliwiają użytkownikowi opuszczenie strefy chronionej od wewnątrz. Jako urządzenia wyjściowe najczęściej używa się przełączników, czujników ruchu lub czytników. Według British Security Industry Association urządzenia wyjściowe

można podzielić je na zwykłe oraz awaryjne (ang. *emergency egress*), przy czym, ze względu na krytyczne znaczenie w wypadku awarii, działanie tych drugich nie powinno zależeć od komponentów systemu (kontrolera systemu, oprogramowania itp.). Jako urządzenie awaryjne często stosuje się tzw. *break glass device*, którego uaktywnienie powoduje odcięcie zasilania w zamku, a tym samym wstrzymanie kontroli dostępu w danym punkcie. Dostęp uzyskany za pomocą tego urządzenia powinien wygenerować stosowne powiadomienie bądź alarm.

W przypadku, gdy system kontroli dostępu nie funkcjonuje odpowiednio (np. z powodu braku zasilania), stosuje się tzw. *break glass device*.

W zależności od potrzeb, oprogramowanie w systemie może być samodzielnym programem zainstalowanym na komputerze osobistym bądź złożonym i bezpiecznym oprogramowaniem zainstalowanym na serwerze. Często opiera się na rozwiązaniach webowych lub mobilnych, umożliwiając dostęp z dowolnego urządzenia

2.2 Przegląd istniejących rozwiązań

Obecny w dzisiejszych czasach postęp technologiczny zapewnia wiele możliwości w kwestii implementacji systemów kontroli dostępu. Prowadzi to do różnorodności rozwiązań, z których każde posiada właściwe dla wykorzystanych metod wady i zalety. Poniżej przedstawiona została zaproponowana w [3] klasyfikacja systemów kontroli dostępu ze względu na metodę uwierzytelniania:

A. System zamków mechanicznych

   W tego typu systemie zamek zawiera układ mechaniczny otwierany przez fizyczny klucz. Jest to wciąż najczęściej stosowana metoda ochrony przestrzeni mieszkalnych, mimo że złamanie takiego zabezpieczenia wymaga jedynie odpowiednich umiejętności manualnych i wykorzystanych narzędzi.

B. System z uwierzytelnieniem przez PIN

   Dostęp do chronionych pomieszczeń w tego rodzaju systemie przydzielany jest na podstawie kodu wprowadzanego za pomocą klawiatury numerycznej podłączonej do mikrokontrolera. Rozwiązanie to cechuje się brakiem konieczności posiadania kosztownego w produkcji unikalnego, fizycznego klucza, który z natury narażony jest na kradzież lub zagubienie [4]. Wadą tego rozwiazania jest ograniczona możliwość zmiany kodu dostępu i podatność na jego wydobycie z pamięci programowalnej.

C. System z uwierzytelnieniem przez RFID

   Rolę nośnika klucza w tego typu rozwiązaniu pełni identyfikator RFID. W przypadku dostatecznego zbliżenia nośnika do anteny umieszczonej w drzwiach lub ich bezpośrednim sąsiedztwie, następuje odczyt i weryfikacja klucza. Sytemy tego typu umożliwiają śledzenie

ruchu użytkowników systemu w czasie rzeczywistym oraz monitorowanie wyposażenia objętego kontrolą dostępu przez umieszczenie w nim identyfikatorów RFID, jak zostało to opisane w [5]. Ze względu na wykorzystanie nieszyfrowanej transmisji bezprzewodowej, przesyłany klucz jest łatwy do przechwycenia.

D. System z uwierzytelnieniem przez dane biometryczne

   Opierając uwierzytelnienie o konwersję unikalnych cech fizycznych użytkownika do postaci numerycznej można całkowicie wyeliminować konieczność istnienia fizycznych kluczy i haseł. Zaletą takiego systemu jest brak możliwości zgubienia lub kradzieży klucza czy zapomnienia hasła. Potencjalnie, cecha ta podnosi poziom bezpieczeństwa obiektu objętego tego typu kontrolą dostępu, jednak rozwiązanie posiada też wady. Wśród przezentowanych w [6] zagrożeń wyróżnić można szczególnie brak tajności danych biometrycznych, co sprawia, że możliwa jest duplikacja danej cechy. Sama cecha biometryczna jest podatna na uszkodzenia mechaniczne, a po utracie nie można jej zastąpić.

E. System z uwierzytelnieniem przez OTP (ang. *One Time Password*, hasło jednorazowe)

   Uwierzytenienie w systemie tego rodzaju wymaga od użytkownika wprowadzenia jednorazowego hasła. Hasło to jest generowane przez serwer i zapamiętywane w systemie, a do użytkownika wysyłane w wiadomości tekstowej. Podczas dokonywania próby dostępu użytkownik podaje otrzymane hasło, które zostaje porównane z hasłem zapamiętanym w systemie. Po dokonaniu porównania jest ono usuwane z systemu. Główną zaletą tego rozwiązania jest znacząco zmniejszona podatność na ataki powtórzeniowe, w których atakujący przechwytuje dane wykorzystane do autoryzacji i używa ich ponownie w celu uzyskania dostępu. Problemem jest jednak konieczność dostępności użytkownikownika w sieci GSM w celu przekazania hasła za pomocą wiadomości SMS.

F. System oparty na kryptograficznym bezpieczeństwie danych

   Bezpieczeństwo tego rozwiązania bazuje na kluczu dostępu znanym wyłącznie przez użytkownika systemu. W pamięci układu zamka zapisywana jest odpowiednia sekwencja, która po transformacji kryptograficznej staje się hasłem dostępu (w tej postaci jest ono przekazywane użytkownikowi). W ten sposób hasło znane jest tylko użytkownikowi co zapewnia, że nawet osoby odpowiedzialne za utrzymanie systemu nie uzyskają dostępu przez pozyskanie hasła. Słabą stroną metody jest oparcie bezpieczeństwa na kluczu kryptograficznym który musi być przechowywany w bezpieczny sposób. **Nie rozumiem o co tu chodzi.**

G. System oparty o komunikację bezprzewodową **TODO: review documentation considering author names** W systemie tego typu przepływ danych odbywa się za pośrednictwem sieci bezprzewodowej. W celu autoryzacji wykorzystywane jest urządzenie mobilne połączone z siecią. W przeciwieństwie do pozostałych wymienionych rozwiązań, rozwiązanie zaproponowane w [7] nie zakłada bezpośredniej komunikacji użytkownika z zamkiem. Urządzenie

mobilne użytkownika generuje zapytanie do serwera, który dokonuje kontroli uprawnień i w przypadku pozytywnej ich weryfikacji, przekazuje decyzję o sukcesie do układu zamka.

Tu może jakieś podsumowanie tych różnych rodzajów jeszcze jakby się dało

2.3 Bezpieczeństwo

Przegląd zagrożeń związanych z bezpieczeństwem - TBD

Projektując system o tak krytycznym znaczeniu jak system kontroli dostępu, należy poświęcić znaczną uwagę zagadnieniom związanym z bezpieczeństwem. Bezpieczeństwo całego systemu nie może zostać osiągnięte, jeśli chociaż jeden z jego elementów nie zostanie odpowiednio zabezpieczony. Należy więc dołożyć wszelkich starań, aby zapewnić odpowiednie mechanizmy bezpieczeństwa we wszystkich warstwach systemu: sprzętowej oraz programowej oraz odpowiednią ochronę danych we wszystkich fazach działania systemu.

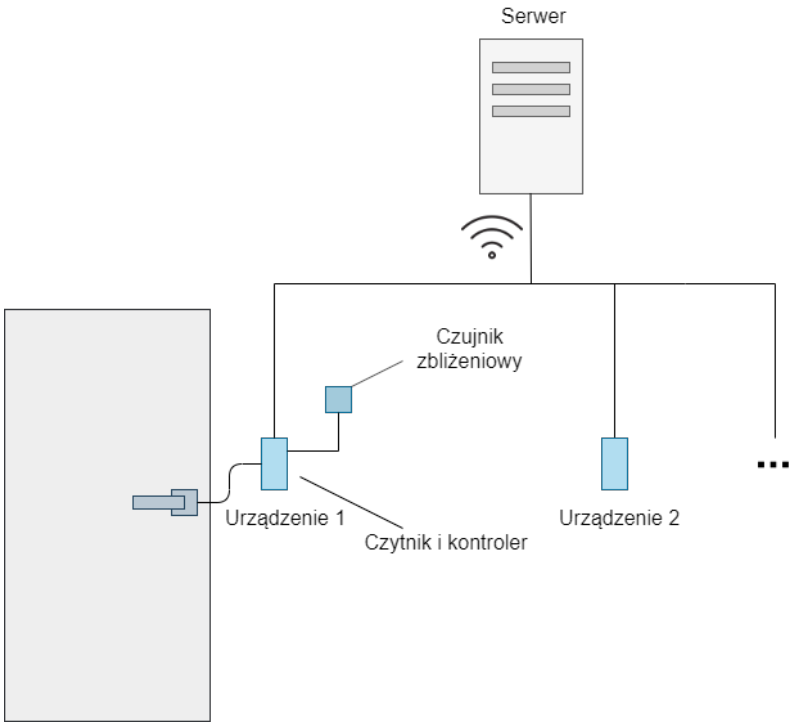
Rozdział 3

Projekt rozwiązania

Niniejszy rozdział przedstawia koncepcję systemu. Definiuje użytkowników oraz wymagania funkcjonalne i pozafunkcjonalne. Przedstawia budowę systemu, opisuje kolejno tworzące system komponenty sprzętowe oraz podsystemy, a także relacje, które między nimi zachodzą. Ponadto prezentuje zakładany sposób działania systemu w kilku najbardziej prawdopodobnych sytuacjach.

3.1 Idea

Idea systemu przedstawiona została na rysnku 3.1.



Rysunek 3.1: Idea systemu

3.2 Użytkownicy

Na potrzeby projektu rozwiązania zidentyfikowano dwóch użytkowników. Ich specyfikacja przedstawiona została w tabeli 3.1.

Tablica 3.1: Użytkownicy systemu

USER_1	**Użytkownik**
Opis	Osoba posiadająca identyfikator RFID, której celem jest uzyskanie dostępu do chronionego zamkiem pomieszczenia

USER_2	**Administrator**
Opis	Osoba posiadająca uprawnienia administracyjne w systemie, mająca dostęp do oprogramowania zarządzającego systemem

3.3 Wymagania

Wymagania funkcjonalne oraz pozafunkcjonalne systemu zostały przedstawione odpowiednio w tabeli 3.2 i 3.3.

Tablica 3.2: Wymagania funkcjonalne

FNRQ_1	**Kontrola dostępu do pomieszczeń**
Opis	Dopuszczanie do pomieszczeń użytkowników posiadających odpowiedni identyfikator i niedopuszczanie użytkowników nieposiadających odpowiedniego identyfikatora

FNRQ_2	**Przeglądanie historii prób dostępu do pomieszczeń**
Opis	Dostęp do listy dokonanych w przeszłości prób dostępu zakończonych zarówno sukcesem, jak i porażką

FNRQ_3	**Dodawanie identyfikatorów**
Opis	Dodawanie identyfikatorów wraz z przyznaniem dostępu do wybranej grupy zamków

Tablica 3.2: Wymagania funkcjonalne, c.d.

FNRQ_4	**Przeglądanie identyfikatorów powiązanych z danym zamkiem**
Opis	Dostęp do listy powiązań między identyfikatorami a zamkami

FNRQ_5	**Blokowanie dostępu do pomieszczeń dla wybranego identyfikatora**
Opis	Manualny wybór opcji czasowego usunięcia powiązania wybranego identyfikatora z wybraną grupą zamków

FNRQ_6	**Dostęp do grupy pomieszczeń za pomocą jednego identyfikatora**
Opis	Ustawienie powiązań identyfikatorów i zamków w taki sposób, aby możliwy był dostęp do grupy zamków za pomocą jednego identyfikatora

FNRQ_7	**Dostęp do pomieszczenia za pomocą grupy identyfikatorów**
Opis	Ustawienie powiązań identyfikatorów i zamków w taki sposób, aby możliwy był dostęp do jednego zamka za pomocą grupy identyfikatorów

FNRQ_8	**Sygnalizacja przyznania lub odmowy dostępu**
Opis	Powiadamianie użytkownika o podjętej przez system decyzji

FNRQ_9	**Sygnalizacja stanu naładowania baterii w układzie zamka**
Opis	Okresowe powiadamianie administratora systemu o bieżącym stanie naładowania baterii

Tablica 3.3: Wymagania pozafunkcjonalne

XXRQ_1	**Długość czasu pracy układu zamka na baterii równa minimum 1 rok**
Opis	Długość czasu pracy układu zamka na baterii powinna wynosić minimum 1 rok

TBD - dodać jakieś wymaganie dotyczące bezpieczeństwa

TBD - dodać wymaganie dotyczące czasu odpowiedzi zamka

3.4 Komponenty

System podzielony został na podsystemy (tabela 3.4) realizujące określone funkcjonalności za pomocą komponentów programowych (tabela 3.6) i istniejące w ramach fizycznych komponentów

sprzętowych (tabela 3.5).

Tablica 3.4: Podsystemy

SSYS_1	**Podsystem sterowania zamkiem**
Opis	Odpowiedzialny za odczyt danych identyfikatora użytkownika oraz przekazanie ich do podsystemu autoryzacji, zarządzanie zasilaniem elementów układu zamka oraz zarządzanie samym zamkiem.
Lokalizacja	HCMP_1 Układ zamka
Komponenty	SCMP_1 Oprogramowanie mikrokontrolera w układzie zamka

SSYS_2	**Podsystem autoryzacji**
Opis	Odpowiedzialny za podjęcie decyzji o przyznaniu bądź odmowie dostępu na podstawie otrzymanych od podsystemu sterowania zamkiem danych. Komunikuje się z podsystemem sterowania zamkiem oraz bazą danych.
Lokalizacja	HCMP_2 Serwer
Komponenty	SCMP_2 Oprogramowanie autoryzujące

SSYS_3	**Podsystem zarządzający**
Opis	Odpowiedzialny za umożliwienie administratorowi systemu wglądu do danych takich jak historia prób dostępu, zbiór identyfikatorów, zamków, oraz powiązań między nimi, a także stan poszczególnych zamków. Dzięki niemu możliwa jest konfiguracja rozpoznawanych przez system identyfikatorów i zamków oraz manualne przyznawanie dostępu poszczególnym identyfikatorom. Nazywany też podsystemem zarządzania.
Lokalizacja	HCMP_2 Serwer
Komponenty	SCMP_3 Oprogramowanie zarządzające, SCMP_4 Baza danych

Tablica 3.5: Komponenty sprzętowe

HCMP_1	**Układ zamka**
Opis	Złożony z następujących subkomponentów: • Mikrokontroler Odpowiada za sterowanie peryferiami, zarządzaniem ich zasilaniem, inicjację i przeprowadzenie bezprzewodowej komunikacji z serwerem i sterowanie samym zamkiem. • Czujnik ruchu Jego jedynym zadaniem jest wykrycie zbliżającego się do zamka użytkownika. • Czytnik RFID Stanowi interfejs pomiędzy użytkownikiem a systemem.
Powiązania	SCMP_1 Oprogramowanie mikrokontrolera w układzie zamka

HCMP_2	**Serwer**
Opis	Komponent na którym zlokalizowane są podsystemy autoryzacji, zarządzania oraz baza danych.
Powiązania	SCMP_2 Oprogramowanie autoryzujące, SCMP_3 Oprogramowanie zarządzające, SCMP_4 Baza danych

Tablica 3.6: Komponenty programowe

SCMP_1	**Oprogramowanie mikrokontrolera w układzie zamka**
Opis	Odpowiedzialne za zarządzanie peryferiami układu, komunikację z serwerem oraz kontrolę zamka.
Powiązania	HCMP_1 Układ zamka

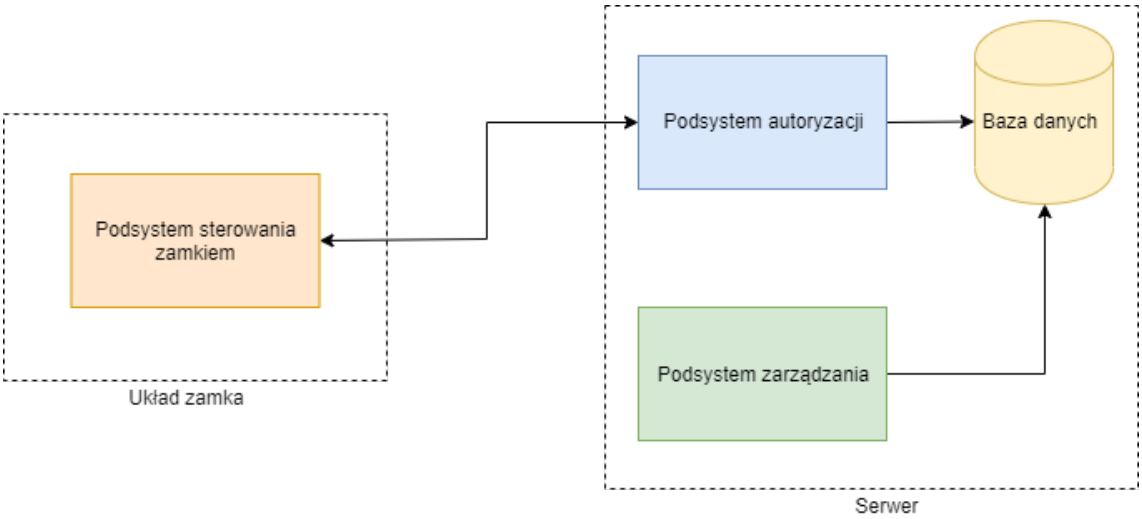
SCMP_2	**Oprogramowanie autoryzujące**
Opis	Zawiera logikę autoryzacyjną. Odbiera dane od układu zamka, komunikuje się z bazą danych w celu podjęcia decyzji o autoryzacji.
Powiązania	HCMP_2 Serwer

SCMP_3	**Oprogramowanie zarządzające**
Opis	Umożliwia zarządzanie istniejącymi identyfikatorami i zamkami oraz dodawanie nowych, przeglądanie historii prób dostępu.
Powiązania	HCMP_2 Serwer

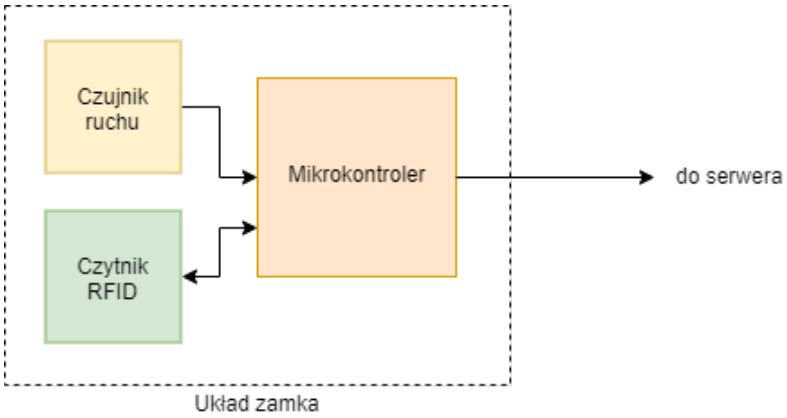
SCMP_4	**Baza danych**
Opis	Przechowuje dane dotyczące poszczególnych identyfikatorów i zamków zarejestrowanych w systemie, powiązań pomiędzy nimi oraz dokonanych w przeszłości prób dostępu zakończonych zarówno sukcesem jak i porażką.
Powiązania	HCMP_2 Serwer

3.5 Projekt architektury

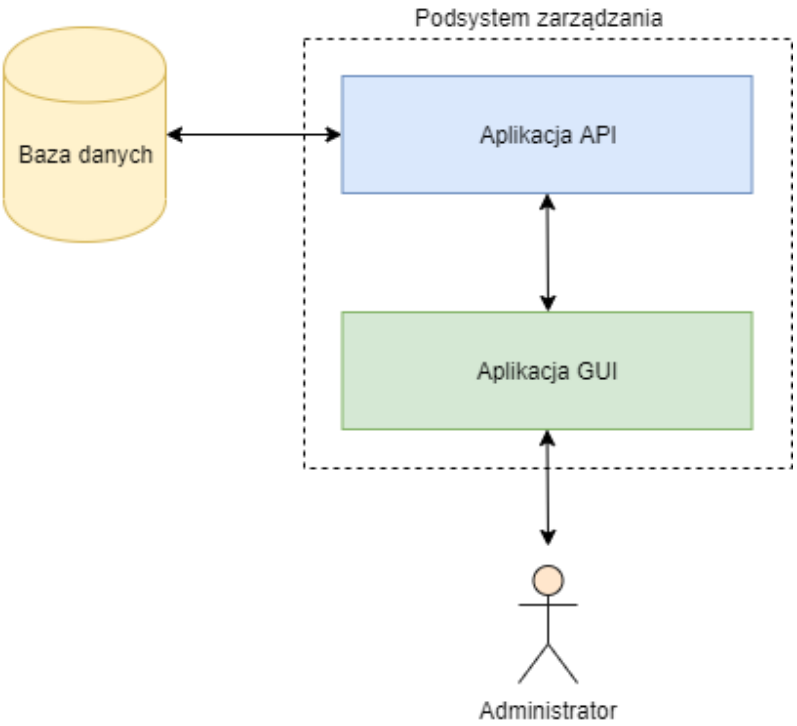
Na architekturę systemu składają się komponenty sprzętowe oraz podsystemy (rysunek 3.2). Rysunek 3.3 przedstawia szczegółową budowę układu zamka. Podsystem zarządzający zbudowany jest z dwóch współpracujących ze sobą aplikacji (rysunek 3.4). **TBD - budowa podsystemu autoryzacyjnego**



Rysunek 3.2: Architektura systemu



Rysunek 3.3: Budowa układu zamka



Rysunek 3.4: Budowa podsystemu zarządzającego

3.6 Sposób działania

Działanie systemu opiera się na współpracy układu zamka z serwerem w celu zapewnienia poprawnej autoryzacji identyfikatorów w zamkach. Funkcjonalność autoryzacji realizowana jest wyłącznie po stronie serwera. Układ zamka, będący urządzeniem klienckim, jest jedynie pośrednikiem przekazującym dane od użytkownika do serwera.

W celu osiągnięcia największej możliwej wydajności energetycznej czytnik RFID oraz mikrokontroler przez większą część czasu pozostają w stanie uśpienia. Zadaniem czujnika ruchu, zasilanego przez cały czas, jest odpowiednio wczesne wykrycie zbliżającego się użytkownika i wybudzenie mikrokontrolera, który z kolei jest odpowiedzialny za zasilenie czytnika RFID oraz nawiązanie połączenia z serwerem. Jeśli operacja nawiązania połączenia przebiegnie pomyślnie, a do czytnika przyłożony zostanie identyfikator, rozpoczyna się proces przekazywania danych odczytanych z identyfikatora do serwera w celu autoryzacji identyfikatora. Należy zwrócić uwagę na niekorzystne z punktu widzenia systemu warunki, które mogą zajść w trakcie procesu:

1. Jeżeli identyfikator nie zostanie przyłożony w przeciągu 10 sek. od momentu wybudzenia czytnika, mikrokontroler ponownie wprowadza czytnik oraz samego siebie w stan uśpienia.
2. Jeżeli połączenie z serwerem nie może zostać nawiązane w czasie $t + 5\ sek.$, gdzie t jest zmiennym czasem upływającym od momentu podjęcia próby nawiązania połączenia z serwerem do momentu zakończenia odczytu danych z identyfikatora, mikrokontroler sygnalizuje użytkownikowi błąd połączenia, po czym wprowadza się w stan uśpienia.

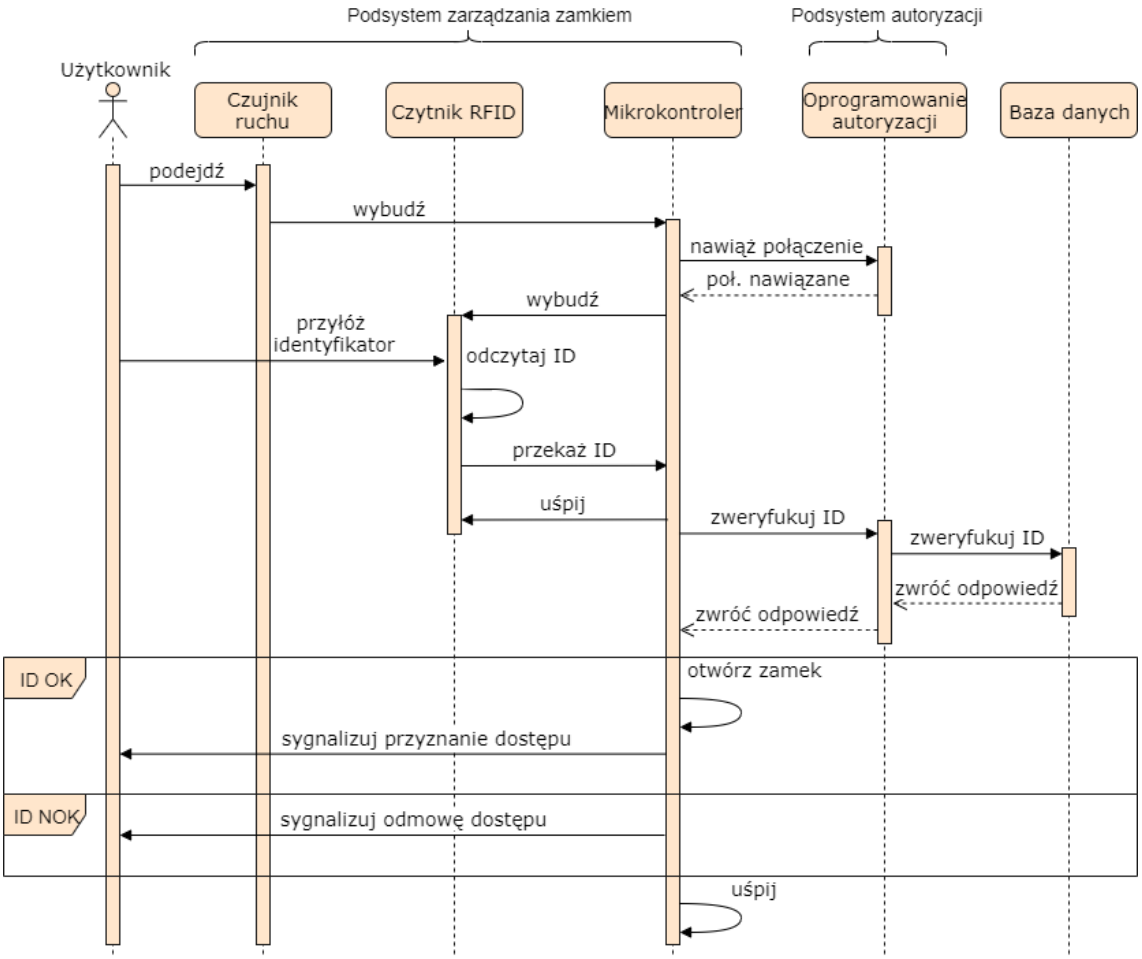
Jeżeli żadna z wymienionych wyżej niekorzystnych sytuacji nie wystąpi i dane zostaną po-

myślnie przesłane do serwera, serwer podejmuje decyzję o przyznaniu bądź odmowie dostępu. Dokonuje tego po wysłaniu zapytania do bazy danych, a następnie wysyła potwierdzenie lub odmowę do mikrokontrolera. Mikrokontroler w sposób wizualny sygnalizuje decyzję użytkownikowi, a jeżeli była ona pomyślna, dodatkowo wysyła sygnał otwierający zamek. Niezależnie od decyzji serwera, wpis o próbie dostępu zostaje zapisany w bazie danych, skąd może być pobrany przez podsystem zarządzający w celu prezentacji danych administratorowi systemu.

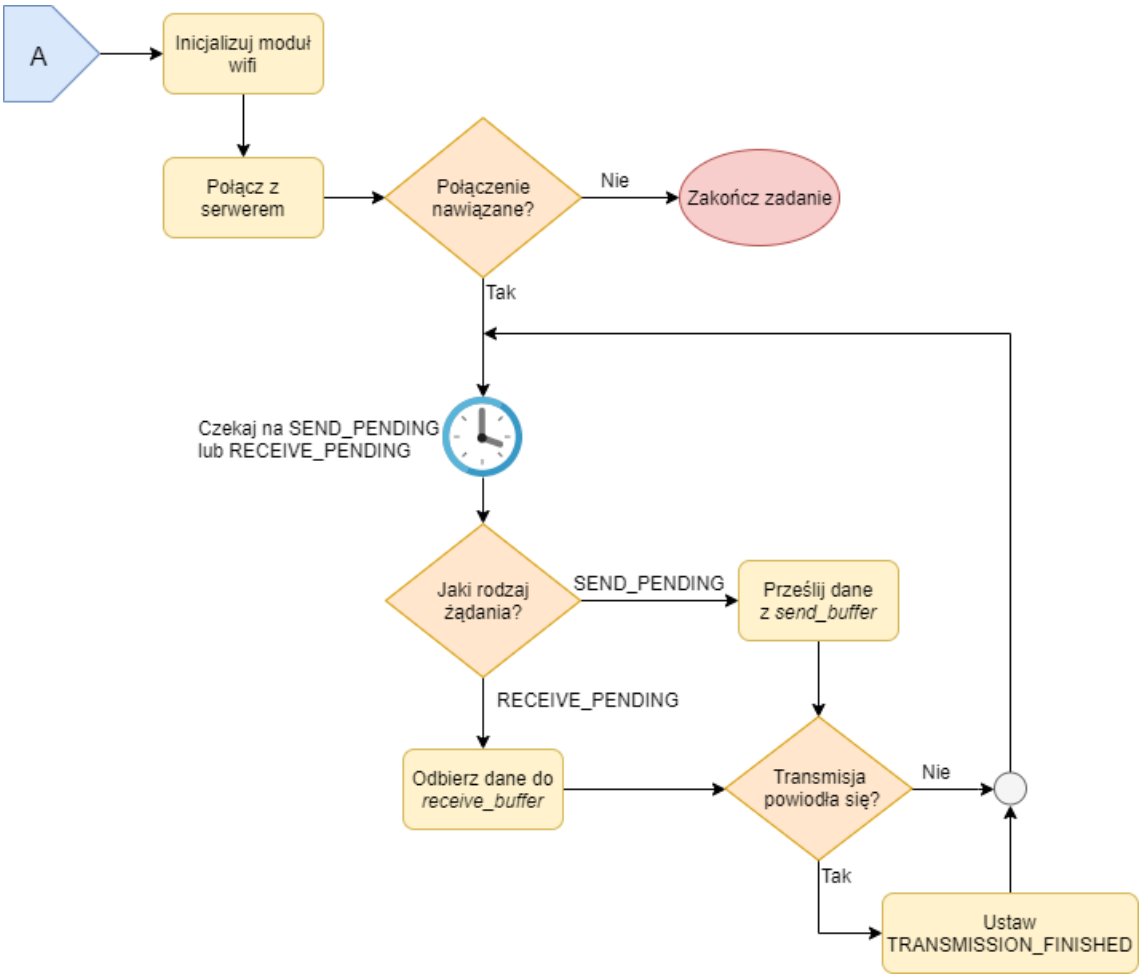
Opisane wyżej mechanizmy zostały szerzej ukazane na diagramach sekwencji. Diagramy 3.5-3.7 dotyczą podsystemu sterowania zamkiem oraz podsystemu autoryzacji, natomiast diagramy 3.8-3.9 dotyczą podsystemu zarządzającego.

Przedstawione sytuacje to kolejno: proces autoryzacji identyfikatora użytkownika w przypadku najbardziej pomyślnego scenariusza (3.5), przepływ sterowania pomiędzy elementami systemu w sytuacji, gdy użytkownik zostanie wykryty, ale identyfikator nie zostanie przyłożony do czytnika w zadanym przedziale czasu (3.6), przepływ sterowania pomiędzy elementami systemu w sytuacji, gdy niemożliwe jest nawiązanie połączenia z serwerem (3.7), przepływ sterowania pomiędzy warstwami podsystemu zarządzającego w sytuacji żądania dostępu do historii prób dostępu przez administratora systemu (3.8) oraz przepływ sterowania pomiędzy warstwami podsystemu zarządzajacego w sytuacji dodania nowego identyfikatora przez administratora systemu (3.9).

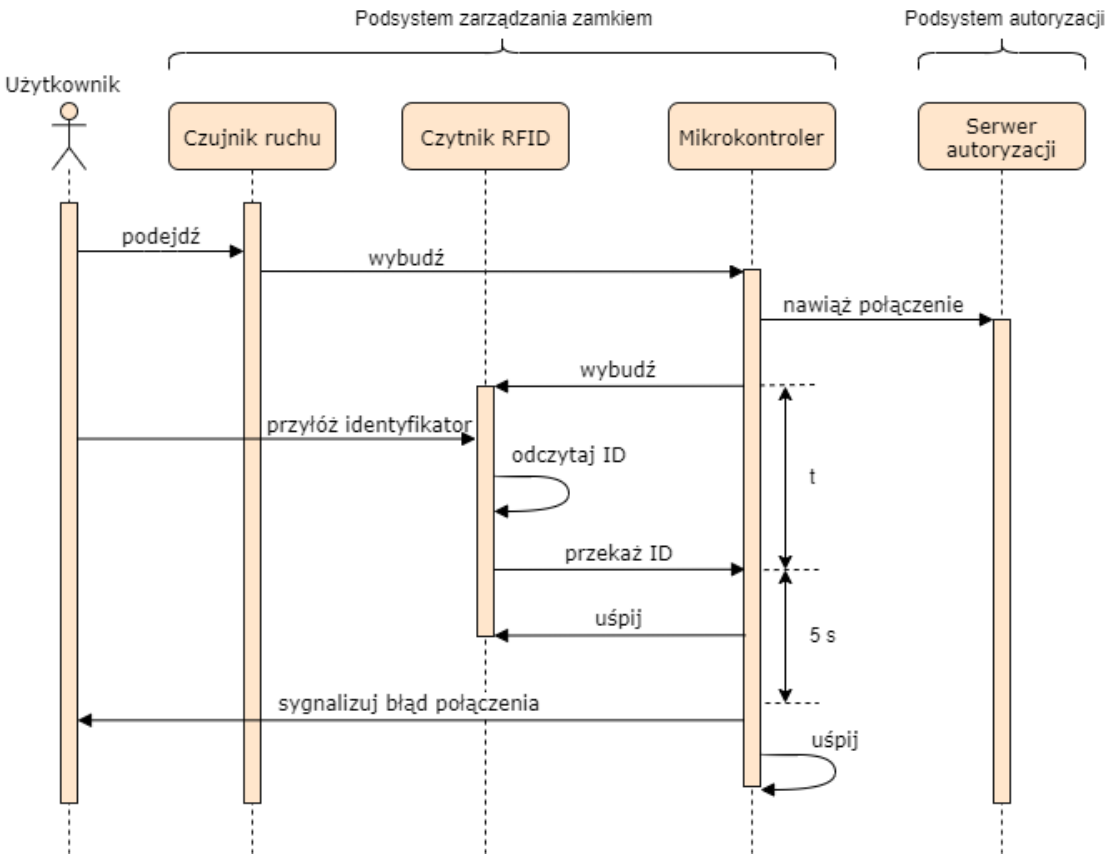
Należy zwrócić uwagę na możliwość wystąpienia również innych sytuacji niekorzystnych, takich jak błąd połączenia z bazą danych lub **co jeszcze?**. Ich obsługa jest pomijalna z punktu widzenia współpracy komponentów systemu, dlatego nie została uwzględniona na diagramach.



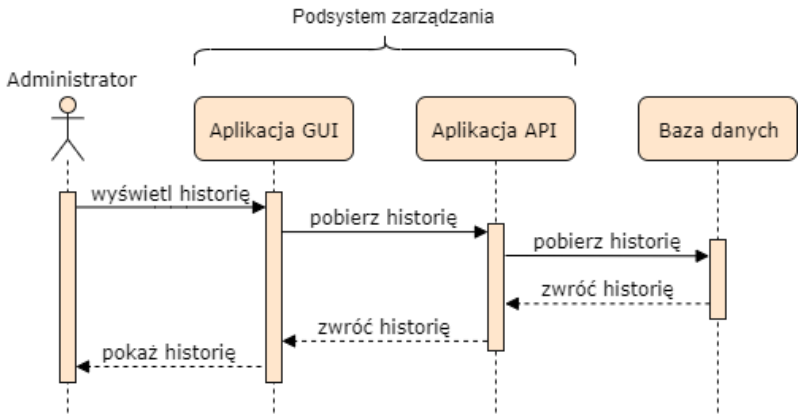
Rysunek 3.5: Przepływ sterowania w procesie autoryzacji identyfikatora



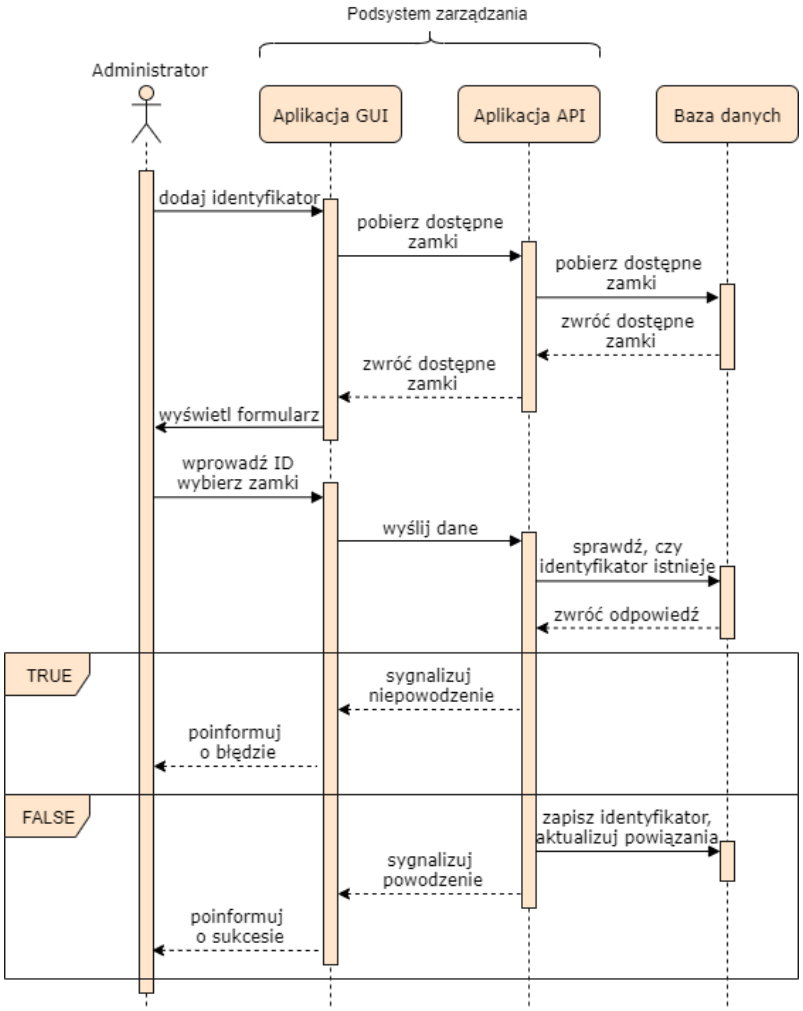
Rysunek 3.6: Przepływ sterowania w sytuacji wykrycia użytkownika nie prezentującego identyfikatora



Rysunek 3.7: Przepływ sterowania w sytuacji braku możliwości nawiązania połączenia z serwerem



Rysunek 3.8: Przepływ sterowania w procesie żądania dostępu do historii prób dostępu



Rysunek 3.9: Przepływ sterowania w procesie dodawania do systemu nowego identyfikatora

Rozdział 4

Układ zamka

Niniejszy rozdział przedstawia zagadnienia związane z procesem implementacji układu zamka. Uzasadnia wybór wykorzystanych w projekcie technologii sprzętowych i programowych, opisuje sposób konfiguracji środowiska deweloperskiego, a także przedstawia wybrane problemy implementacyjne.

4.1 Wykorzystane technologie

4.1.1 Mikrokontroler

Istotnym problemem natury implementacyjnej jest wybór odpowiedniego mikrokontrolera, który spełni wymagania dotyczące wydajności energetycznej oraz bezpieczeństwa systemu, jednocześnie zapewniając możliwość obsługi kilku układów peryferyjnych i wsparcie dla komunikacji bezprzewodowej, a także wygodne i elastyczne środowisko deweloperskie.

Spośród wielu rozwiązań dostępnych na rynku wybrany został mikrokontroler ESP32. Oprócz wszystkich wymienionych wyżej cech, charakteryzuje się on możliwością przetwarzania równoległego za pomocą dwóch fizycznych rdzeni, co otwiera nowe możliwości w kwestii projektowania i implementacji oprogramowania, szczególnie w przypadku istnienia rygorystycznych wymagań dotyczących czasu odpowiedzi układu. Zapewnia sprzętowe wsparcie dla algorytmów kryptograficznych oraz możliwość zaawansowanej kontroli pracy podzespołów w zakresie zasilania, a także umożliwia komunikację z peryferiami za pośrednictwem programowalnych wejść/wyjść GPIO (ang. *General-Purpose Input/Output*). **Napisać jakie ma currenty**

Oprogramowanie dla mikrokontrolera ESP32 może być wytwarzane w środowisku Arduino IDE lub z wykorzystaniem oficjalnej platformy programistycznej Espressif IoT Development Framework (ESP-IDF) firmy Espressif Systems. ESP-IDF opiera się na systemie FreeRTOS (ang. *Free Real-Time Operating System*, system operacyjny czasu rzeczywistego), w wersji 8.2.0, port Xtensa [8]. Ze względu na chęć uzyskania pełniejszej kontroli nad mikrokontrolerem zdecydowano się na użycie w projekcie platformy ESP-IDF.

4.1.2 Czytnik RFID

Głównym kryterium wyboru czytnika identyfikatorów RFID jest prostota integracji z mikrokontrolerem. Ze względu na fakt, iż jego zasilaniem zarządzać będzie mikrokontroler, niski pobór mocy nie stanowi kryterium przesądzającego. Wybrano układ MFRC522 produkowany przez firmę NXP Semiconductors. Układ obsługuje urządzenia o częstotliwości nośnej 13,56 MHz oraz umożliwia komunikację poprzez protokół SPI (ang. *Serial Peripheral Interface*).

4.1.3 Czujnik ruchu

Czujnik ruchu posiada krytyczne znaczenie w układzie zamka. Jako jedyny z podzespołów musi być zasilany przez cały czas pracy układu. Ze względu na wymaganie wykorzystania zasilania bateryjnego w układzie istotną cechą jest jak najniższy pobór mocy w stanie spoczynku. Wybranym rozwiązaniem jest czujnik PIR (ang. *Passive Infra Red*, pasywny czujnik podczerwieni), model HC-SR501. Wykrycie obiektu sygnalizowane jest stanem wysokim na cyfrowym wyjściu. Czujnik HC-SR501 wyposażony jest w dwa potencjometry umożliwiające regulację czasu trwania stanu wysokiego po wykryciu obiektu oraz czułości czujnika. Układ posiada także zworkę, którą można samodzielnie zlutować i w ten sposób wybrać jeden z dwóch możliwych trybów:

- Tryb *repeatable* - po wykryciu obiektu stan wysoki utrzymywany jest przez określony przez potencjometr czas,
- Tryb *non-repeatable* - stan wysoki po wykryciu obiektu utrzymywany jest przez cały czas występowania ruchu. Po zatrzymaniu ruchu stan wysoki utrzymywany jest przez określony przez potencjometr czas.

Dzięki temu możliwe jest dobranie parametrów czujnika w taki sposób, aby użytkownik był wykrywany w odpowiednim momencie, co przekłada się na bardziej optymalne działanie układu.

Napisać jak u nas są te parametry ustawione

Napisać jaki ma zasięg, sprzeczne źródła :/

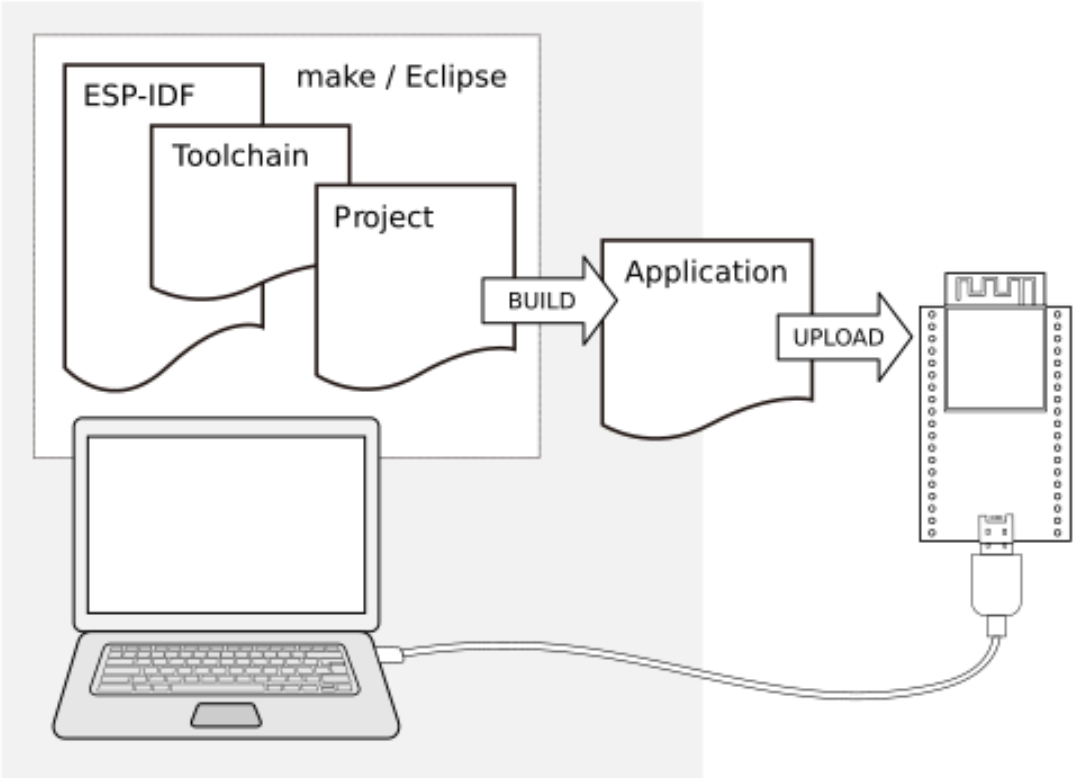
TBD - schemat jak sa polaczone pinami te wszystkie komponenty

4.2 Konfiguracja środowiska

Platforma programistyczna ESP-IDF zapewnia zestaw narzędzi (ang. *toolchain*) umożliwiających wytwarzanie oprogramowania dla mikrokontrolera ESP32. Pobranie ESP-IDF sprowadza się do sklonowania repozytorium. Platforma ESP-IDF nie jest częścią projektu, lecz samodzielnym modułem linkowanym do projektu za pomocą zmiennej środowiskowej.

Rysunek 4.1 [9] przedstawia komponenty potrzebne do wytwarzania aplikacji dla ESP32.

Konfiguracja, budowanie i **flashowanie jak to powiedziec po polsku** aplikacji odbywa się za pomocą dedykowanego skryptu dostarczanego przez platformę.



Rysunek 4.1: Wytwarzanie oprogramowania dla ESP32

4.3 Oprogramowanie mikrokontrolera

4.3.1 Komponenty

ESP-IDF umożliwia podział aplikacji na komponenty oraz ich łatwą konfigurację w procesie kompilacji za pomocą tekstowego menu. Jak wyjaśnia [10], komponenty to modularne i samodzielne fragmenty kodu kompilowane do bibliotek statycznych i linkowane do aplikacji. Niektóre z nich są dostarczane przez platformę ESP-IDF. Istnieje też możliwość tworzenia własnych komponentów.

Kod źródłowy oprogramowania mikrokontrolera został podzielony na cztery komponenty:

1. Komponent główny (*main*),
2. Komponent RFID,
3. Komponent WiFi,
4. Komponent sterujący (*flow-controller*).

Napisac ktore rzeczy sa konfigurowalne przez menuconfig

4.3.2 Współbieżność

Jak opisuje [8], oryginalny system FreeRTOS został zaprojektowany z myślą o przetwarzaniu za pomocą jednego rdzenia. ESP32 posiada jednak dwa rdzenie o wspólnej pamięci, co pozwala na przemienne wykonywanie zadań na obu rdzeniach. Koncepcja zadań (ang. *tasks*) istniejąca w oryginalnym FreeRTOS w ESP-IDF została zmodyfikowana na potrzeby wsparcia mikrokontrolera o dwóch rdzeniach. Każde zadanie ma określony priorytet, na podstawie którego algorytm planowania ustala kolejność wykonywania. Dzieje się to indywidualnie dla każdego rdzenia, lecz lista zadań gotowych do wykonania jest pomiędzy nimi współdzielona.

Aby maksymalnie wykorzystać możliwości platformy ESP-IDF w kwestii współbieżności jednocześnie unikając nadmiernego skompilowania kodu, zdecydowano się na wykorzystanie w ramach aplikacji dwóch zadań. Należy nadmienić, że przez większą część czasu mikrokontroler nie wykonuje operacji, których zrównoleglenie byłoby korzystne. Jedyną sytuacją, w której warto zastosować przetwarzanie współbieżne jest nawiązywanie połączenia z serwerem oraz odczyt danych z czytnika RFID. Obie czynności mogą okazać się czasochłonne. W aplikacji jednowątkowej, odczyt danych z czytnika RFID musiałby zakończyć się przed rozpoczęciem nawiązywania połączenia, lub odwrotnie. Jeśli zastosujemy zapewniany przez ESP-IDF mechanizm zadań, możemy wykonywać obie operacje jednocześnie.

Do synchronizacji zadań wykorzystany został mechanizm tzw. *event group* (grupy zdarzeń) dostarczany przez system FreeRTOS. Zakłada on wykorzystanie przez współpracujące zadania współdzielonej grupy zdarzeń, składającej się z zestawu flag, oraz udostępnia funkcje do operowania na niej. Możliwe operacje to ustawianie w stan wysoki lub niski danej flagi oraz oczekiwanie na stan wysoki flagi. Korzystny z perspektywy synchronizacji wątków jest fakt, że oczekiwanie na dane zdarzenie blokuje aktualnie wykonywane zadanie, umożliwiając oddanie sterowania innym oczekującym zadaniom.

Tutaj jakiś diagram pokazujący komunikację między taskami czy coś

4.3.3 Stan głębokiego uśpienia

ESP32 oferuje efektywną i elastyczną technologię zarządzania energią. Dokument [11] wymienia pięć dostępnych stanów energetycznych:

1. *Active mode* Aktywne CPU wraz z układem radiowym.
2. *Modem-sleep mode* Aktywne CPU, układ radiowy wyłączony.
3. *Light-sleep mode* Uśpione CPU. Pamięć i peryferia RTC (ang. *Real-Time Clock*, zegar czasu rzeczywistego) wraz z koprocesorem ULP (ang. *Ultra Low Power co-processor*) są aktywne. Jakiekolwiek zdarzenia wybudzające doprowadzą do wybudzenia układu.
4. *Deep-sleep mode* Tylko pamięć i peryferia RTC pozostają zasilone. Dane dotyczące połączeń WiFi i Bluetooth zostają przechowane w pamięci RTC. Opcjonalnie dostępny jest koprocesor ULP.

5. *Hibernation mode* Wewnętrzny rezonator kwarcowy wraz z koprocesorem ULP zostają wyłączone. Również pamięć RTC jest wyłączona. Wybudzenie możliwe jest tylko poprzez timer RTC lub wejścia RTC GPIO.

Można dodać jakie są zużycia energii tutaj

Z punktu widzenia wymagań dotyczących poboru energii niezwykle przydatnym trybem jest tryb głębokiego uśpienia (ang. *deep-sleep mode*). Dzięki jego wykorzystaniu możliwe jest przebywanie mikrokontrolera w trybie niskiego zużycia energii przez większą część czasu. Ze względu na cyfrowy charakter wyjścia czujnika ruchu, jako sposób wybudzenia mikrokontrolera skonfigurowano tryb EXT0 (ang. *External Wakeup 0*, wybudzenie zewnętrzne) umożliwiający wykorzystanie modułu RTC IO (ang. *Real-Time Clock Input Output*) w celu zainicjowania wybudzenia. Inne możliwe źródła wybudzenia to m.in. zegar RTC (ang. *timer*) czy przerwanie wbudowanego czujnika dotykowego (ang. *touch pad*). Konfiguracja trybu wybudzania odbywa się programowo jako jedna z pierwszych operacji aplikacji.

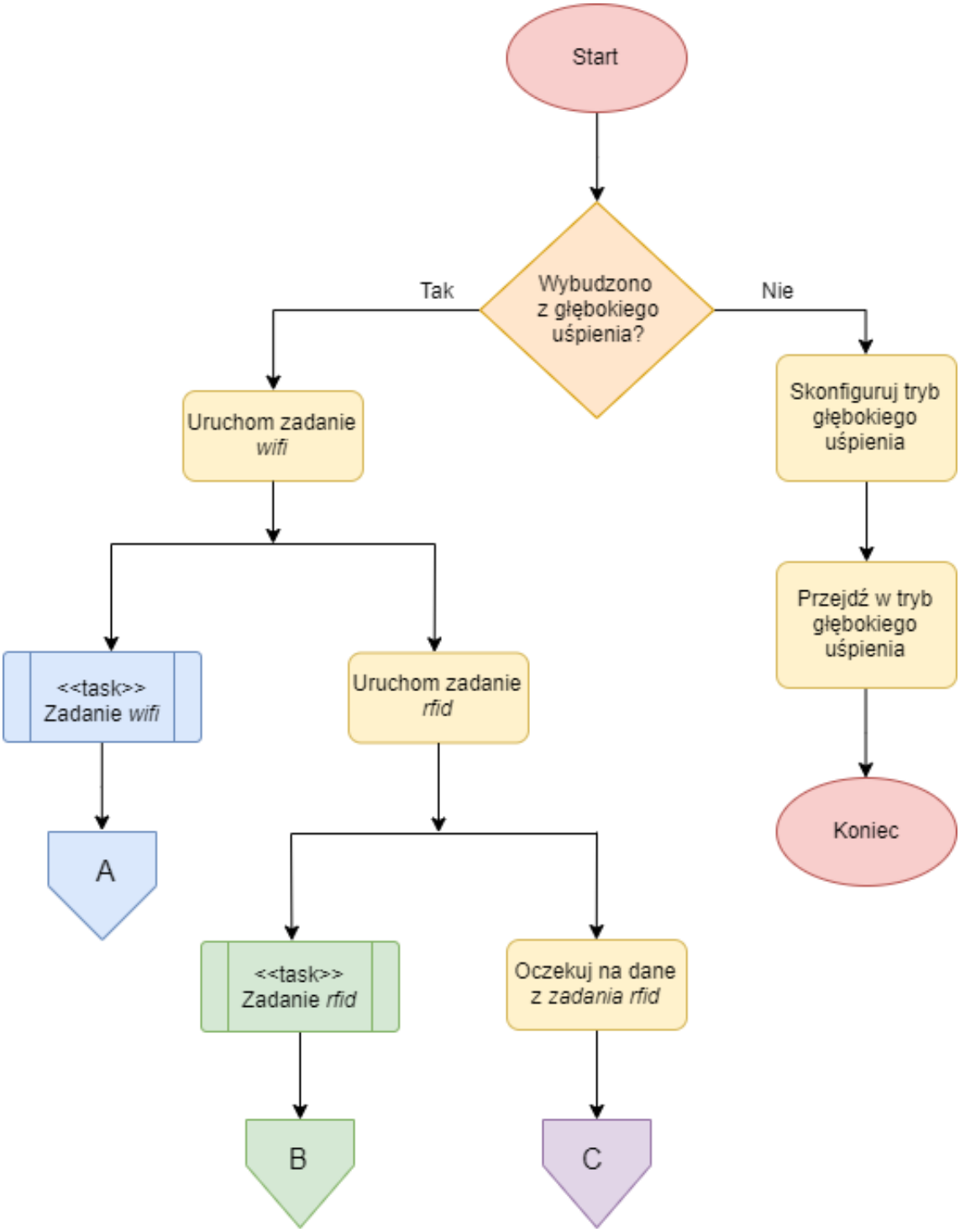
4.3.4 Biblioteka do zarządzania czytnikiem RFID

Do zarządzania czytnikiem MFRC522 wykorzystano zewnętrzną bibliotekę[1].

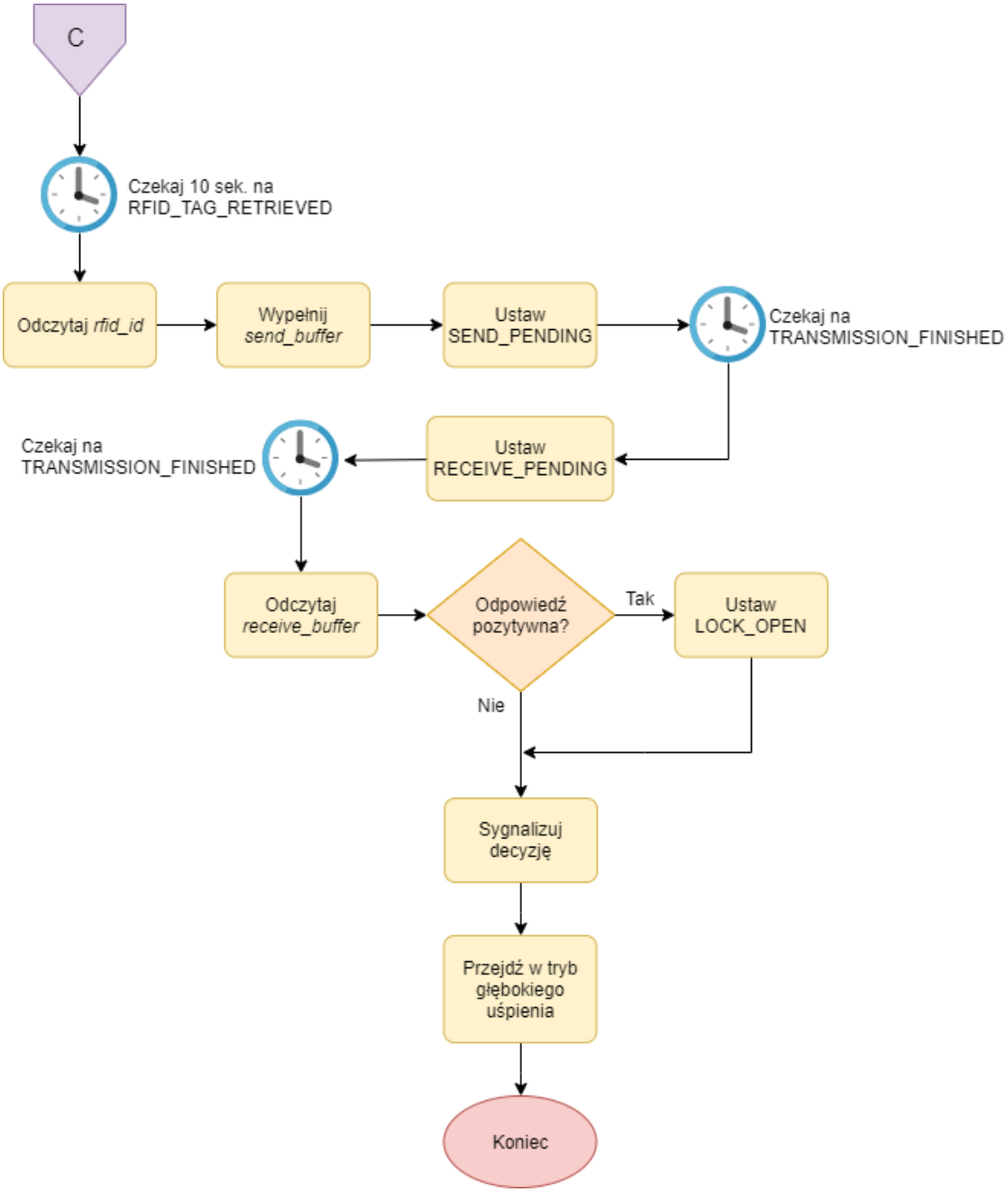
4.3.5 Przepływ sterowania

Przepływ sterowania w oprogramowaniu mikrokontrolera ukazują rysunki 4.2 i 4.3. Na rysunkach 4.4 i 4.5 przedstawiono przepływ sterowania w zadaniach *wifi* i *rfid*.

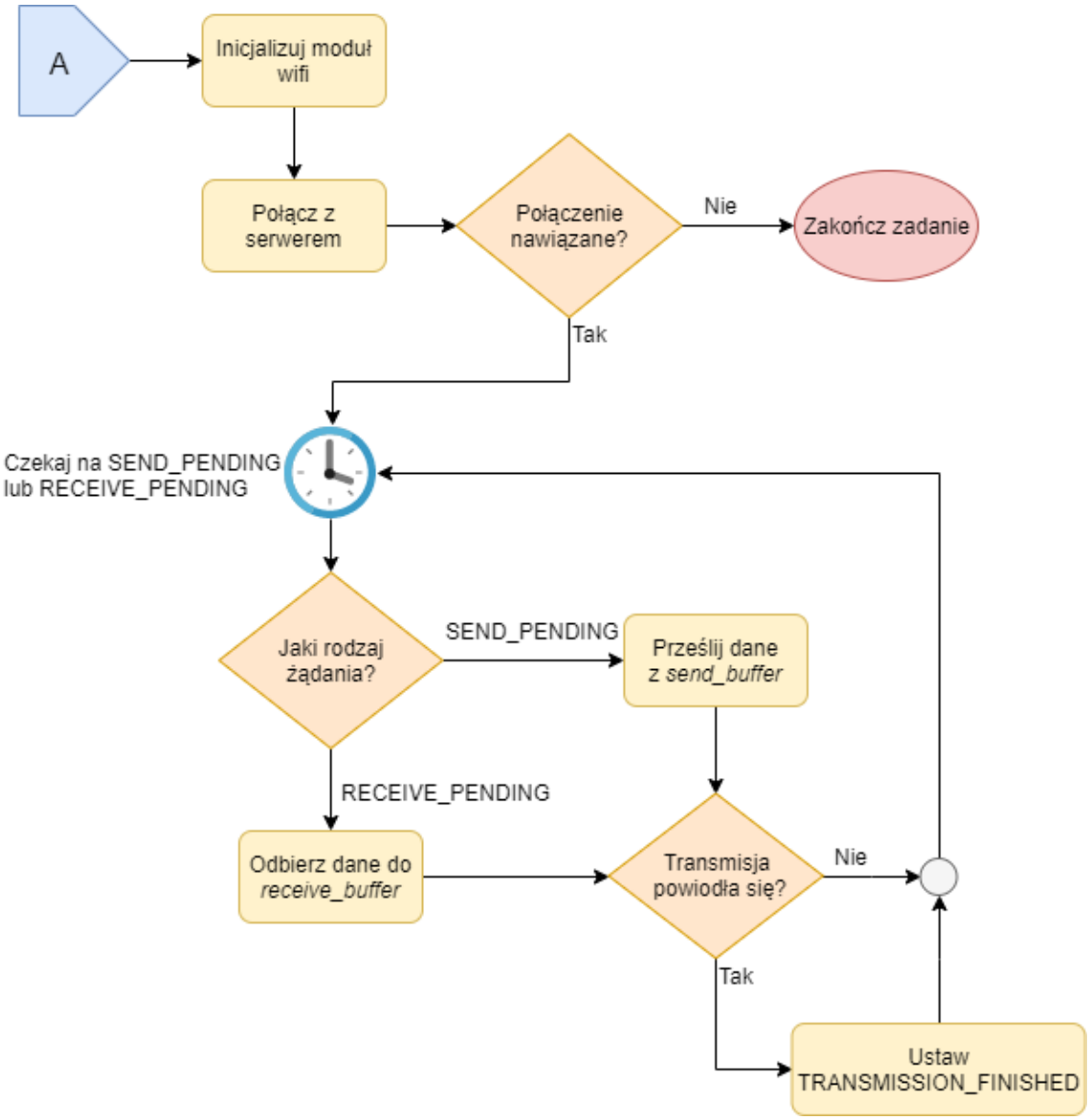
[1] Alija Bobija, *C library for interfacing ESP32 with MFRC522 RFID card reader*, `https://github.com/abobija/esp-idf-rc522` (rewizja 557af67)



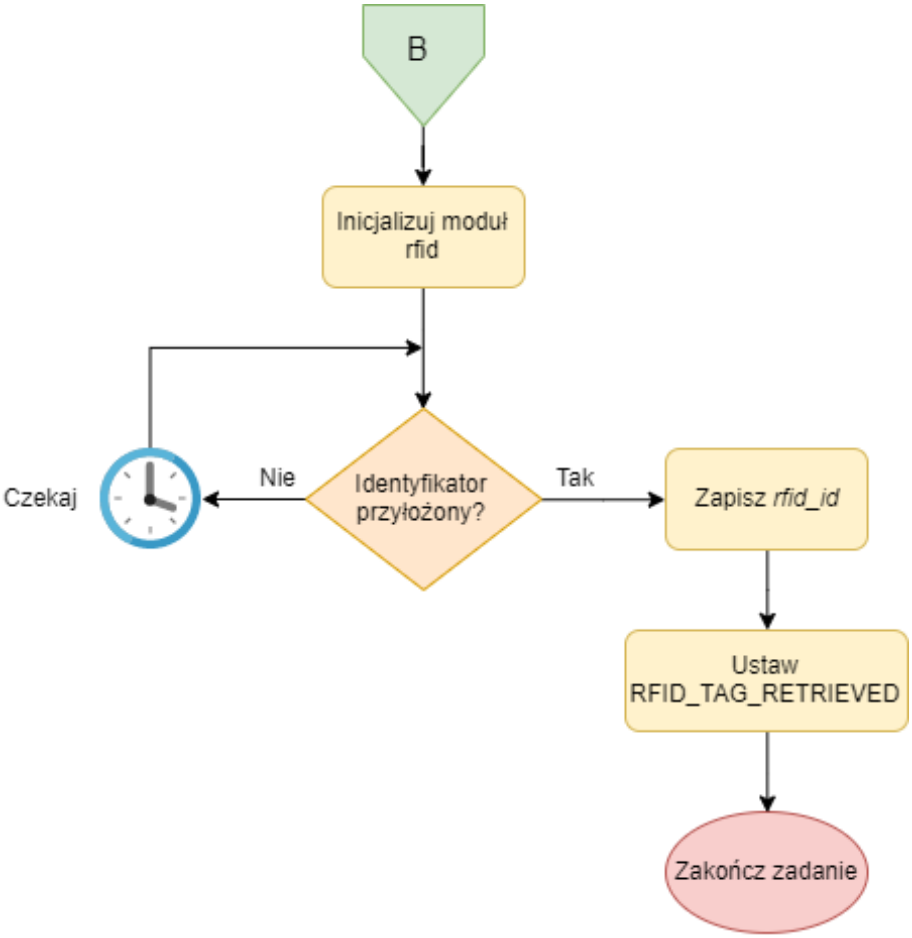
Rysunek 4.2: Schemat blokowy przepływu sterowania oprogramowania mikrokontrolera



Rysunek 4.3: Schemat blokowy przepływu sterowania oprogramowania mikrokontrolera, c.d.



Rysunek 4.4: Schemat blokowy przepływu sterowania zadania obsługującego komunikację z serwerem



Rysunek 4.5: Schemat blokowy przepływu sterowania zadania obsługującego czytnik RFID

Rozdział 5

Serwer

Rozdział 6

Podsumowanie

Niniejszy rozdział dokonuje podsumowania rezultatów prowadzonej pracy. Porównuje szacunki dotyczące poboru mocy z rzeczywistym poborem. Ponadto dokonuje oceny bezpieczeństwa oraz responsywności systemu.

W ramach niniejszej pracy stworzono funkcjonalny prototyp systemu, który po odpowiednich modyfikacjach miałby szansę efektywnej pracy w odpowiednim środowisku.

Dzięki wykorzystaniu zdalnego serwera do przeprowadzenia procesu uwierzytelniania system zapewnia większą elastyczność i łatwość zarządzania niż alternatywne systemy wykorzystujące zamki pracujące w sposób autonomiczny.

Rozwiązanie cechuje się wygodą montażu, ponieważ nie wymaga przewodów zasilających i komunikacyjnych prowadzonych w ścianach budynków. Przy wdrażaniu rozwiązania nie jest konieczna modyfikacja istniejącej infrastruktury budynku, z wyjątkiem wymiany samych zamków. System nie wymaga żadnych dodatkowych komponentów sprzętowych poza zamkami i serwerem.

Wydajność energetyczna podsystemu sterowania zamkiem została osiągnięta przez zarządzanie zasilaniem jego peryferiów oraz kontrolę stanu zasilania mikrokontrolera w celu minimalizacji poboru mocy i maksymalizacji czasu pracy na zasilaniu bateryjnym.

Bezpieczeństwo systemu na wielu poziomach zapewnia wykorzystanie mechanizmów takich jak TLS w warstwie komunikacji pomiędzy zamkiem a serwerem czy szyfrowanie pamięci Flash w warstwie operacji na danych w mikroprocesorze w układzie zamka.

Wydajność energetyczna - TBD

Responsywność - TBD

Bezpieczeństwo - TBD

6.1 Możliwe rozszerzenia

W ramach niniejszej pracy stworzony został prototyp rozwiązania. Niektóre planowane funkcjonalności nie zostały zaimplementowane ze względu na ograniczenia czasowe i budżetowe.

Pozostawiono jednak możliwość rozbudowy systemu. Poniżej przedstawiono kilka problemów, których system w obecnym stanie nie adresuje, wraz z możliwymi rozwiązaniami.

6.1.1 Mechanizm wyjścia

Opisywane rozwiązanie nie obejmuje implementacji mechanizmu opuszczenia strefy chronionej systemem kontroli dostępu. W zależności od potrzeb końcowego użytkownika, możliwe rozwiązanie to montaż dodatkowego czytnika po przeciwnej stronie drzwi i połączenie go z kontrolerem wejścia w przypadku gdy wymagana jest obustronna kontrola dostępu bądź zastosowanie przycisku którego naciśnięcie powoduje zwolnienie zamka w przypadku gdy wymagana jest tylko kontrola wejścia do chronionego obszaru.

6.1.2 Obsługa większej liczby zamków

W celu umożliwienia obsługi przez system liczby zamków przekraczającej 1, wystarczająca byłaby modyfikacja podsystemu autoryzacji w taki sposób, aby mógł on obsługiwać równoległe żądania od klientów.

6.1.3 Wygodna konfiguracja parametrów sieci

W obecnej implementacji dane dostępu do sieci (nazwa sieci oraz hasło) zostały zagnieżdżone w oprogramowaniu konrolera. Zmniejsza to elastyczność konfiguracji urządzenia, wymagając jego przeprogramowania za każdym razem gdy zmianie ulegnie nazwa lub klucz dostępu do sieci.

Możliwym rozwiązaniem tego problemu byłaby implementacja trybu konfiguracji. Tryb ten powodowałby przejście kontrolera w tryb Access Point przy zachowaniu dwóch warunków: (1) nastąpiło uruchomienie, a nie wybudzenie z trybu głębokiego uśpienia oraz (2) na określonym wejściu pojawiło się napięcie. Przejście kontrolera w tryb Access Point umożliwiłoby udostępnienie prostego interfejsu webowego, za pomocą którego administrator systemu mógłby wprowadzić niezbędne do działania dane, takie jak nazwa sieci, hasło, a także adres IP i numer portu serwera autoryzacji. Aby zachować wysoki poziom bezpieczeństwa, komunikacja pomiędzy urządzeniem administratora i kontrolerem powinna odbywać się przy wykorzystaniu protokołu TLS.

6.1.4 Sygnalizacja stanu baterii

W obecnym stanie system nie implementuje mechanizmów informowania serwera o swoim stanie energetycznym. Sygnalizacja stanu baterii umożliwiłaby administratorowi systemu bieżące monitorowanie wszystkich zamków objętych systemem oraz szybką reakcję w przypadku, gdy baterie wymagałyby wymiany. Modyfikacja ta wymagałaby uzyskania dostępu do danych na temat naładowania baterii przez mikrokontroler oraz przesyłania ich okresowo do serwera, najlepiej w momentach, gdy jest on już wybudzony z powodu wykrycia ruchu w jego otoczeniu.

6.1.5 Obsługa kont użytkowników w podsystemie zarządzania

Podsystem zarządzania nie implementuje mechanizmu dostępu do zasobów, co czyni system mniej bezpiecznym. W końcowym produkcie należałoby rozszerzyć go o możliwość tworzenia kont użytkowników i przypisywania im określonych uprawnień co do odczytu i zapisu danych.

Bibliografia

[1] Polski Komitet Normalizacyjny, *Technika informatyczna. Zabezpieczenia w systemach informatycznych. Terminologia*, Marzec 2002.

[2] British Security Industry Association, *A specifier's guide to access control systems*, Kwiecień 2016.

[3] R. S. Divya and M. Mathew, "Survey on various door lock access control mechanisms," in *2017 International Conference on circuits Power and Computing Technologies [ICCPCT]*, IEEE, 2017.

[4] D. C. Poirier and S. R. Vishnubhotla, "A microprocessor-controlled door lock system," in *IEEE Transactions on Consumer Electronics*, vol. 36, pp. 129 – 136, IEEE, 1990.

[5] A. ur Rehman, A. Z. Abbasi, and Z. A. Shaikh, "Building a smart university using rfid technology," in *2008 International Conference on Computer Science and Software Engineering*, IEEE, 2008.

[6] M. Faundez-Zanuy, "On the vulnerability of biometric security systems," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, pp. 3–8, Czerwiec 2004.

[7] M. Ahtsham, H. Y. Yan, and U. Ali, "Iot based door lock surveillance system using cryptographic algorithms," in *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, IEEE, 2019.

[8] E. Systems, "Esp-idf docs api guides freertos smp changes." `https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/freertos-smp.html`, 2019. (rewizja 93a8603c).

[9] E. Systems, "Esp-idf docs get started." `https://docs.espressif.com/projects/esp-idf/en/latest/get-started/index.html`, 2019. (rewizja 93a8603c).

[10] E. Systems, "Esp-idf docs api guides build system." `https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/build-system.html?highlight=binary?highlight=binary#build-system`, 2019. (rewizja 93a8603c).

[11] Espressif Systems, *ESP32 Series Datasheet*, 2019. Version 3.2.

Dodatki

Dodatek A

Uruchomienie projektu

TBD