

UNIwersytet Pedagogiczny im. KEN w KRAKOWIE  
Instytut Informatyki



# SPRAWOZDANIE projekt inżynierski

III rok Informatyka studia niestacjonarne

9 czerwca 2021

## Spis treści

<b>1</b>	<b>Temat projektu</b>	<b>3</b>
<b>2</b>	<b>Zespół projektowy</b>	<b>3</b>
<b>3</b>	<b>Cel i zakres projektu</b>	<b>3</b>
<b>4</b>	<b>Założenia projektowe</b>	<b>3</b>
4.1	Oprogramowanie . . . . .	3
4.2	Komponenty sprzętowe . . . . .	4
<b>5</b>	<b>Opis rozwiązania technicznego</b>	<b>5</b>
5.1	Projekt koncepcyjny . . . . .	5
5.2	Realizacja techniczna . . . . .	6
5.2.1	Schemat podłączeń . . . . .	6
5.2.2	Opis schematu podłączeń . . . . .	7
5.3	Dokumentacja . . . . .	9
<b>6</b>	<b>Testy techniczne</b>	<b>10</b>
6.0.1	Opis testu . . . . .	11
<b>7</b>	<b>Podsumowanie i wnioski</b>	<b>11</b>
<b>8</b>	<b>Załączniki</b>	<b>12</b>

# 1 Temat projektu

*Zamek elektroniczny z wielopoziomą kontrolą dostępu.*

# 2 Zespół projektowy

*Martyna Kłosek*

*e-mail: martyna.klosek@student.up.krakow.pl*

*Jakub Mazurczyk*

*e-mail: jakub.mazurczyk@student.up.krakow.pl*

# 3 Cel i zakres projektu

Celem projektu było zaprojektowanie, zmontowanie i zaprogramowanie układu zamka elektronicznego z wielopoziomą kontrolą dostępu.

Zamek powinien posiadać trójstopniową kontrolę dostępu w trybie kaskadowym - użytkownik dopiero po pomyślnym rozbrojeniu jednej blokady może przejść do kolejnej.

Poziom blokad i zabezpieczeń oraz postęp w ich wyłączaniu jest sygnalizowany poprzez zapalenie się odpowiedniej diody LED oraz wyświetlenie odpowiedniego komunikatu informacyjnego.

System powinien obsługiwać klawiaturę, wyświetlacz ciekłokrystaliczny i symulować mechanizm zamka poprzez sterowanie serwomechanizmem.

Wybranymi przez nas rodzajami zabezpieczeń zostały:

1. Pomiar temperatury ciała.
2. Odczyt breloka.
3. Wpisanie poprawnego kodu.

# 4 Założenia projektowe

Projekt opiera się o technologię mikrokontrolerów Arduino. Wykorzystywaną przez nas płytką było ArduinoUno z 8-bitowym mikrokontrolerem AVR ATmega328.

Dokładny opis wykorzystanych części, czujników i modułów znajduje się w rozdziale 4.2.

Płytkę programowaliśmy za pomocą dedykowanego środowiska ArduinoIDE, które posiada ogromną ilość wbudowanych bibliotek, oferuje dostęp do przykładów i tutoriali co ułatwia pracę z mikrokontrolerem. Za pomocą ArduinoIDE wgrywaliśmy również nasz program na płytkę mikrokontrolera.

## 4.1 Oprogramowanie

W ciągu całego cyklu życia projektu musieliśmy korzystać z różnego rodzaju oprogramowania. Podstawowym wykorzystywanym przez nas oprogramowaniem było środowisko **ArduinoIDE** - wieloplatformowa aplikacja służąca do pisania kodu w C i C++ i przesyłania programów na płyty kompatybilne z Arduino.

Do zaprogramowania naszego mikrokontrolera posługiwaliśmy się językiem C, który dzięki domyślnie zawartej w Arduino bibliotece, był bardziej przyjazny.

Drugim kluczowym oprogramowaniem była aplikacja [Fritzing](#), będąca open-source’owym oprogramowaniem. Pozwala ona projektować urządzenia elektroniczne, tworzyć dla nich prototypy na płytkach stykowych, a następnie na podstawie prototypów rysować i edytować schematy elektroniczne. Aplikacja jest projektem akademickim, który powstał na uniwersytecie w Poczdamie jako wolne oprogramowanie, jest zatem darmowa.

W trakcie programowania zamka korzystaliśmy z kilku bibliotek, które umożliwiły i ułatwiły obsługę modułów i sterowanie nimi.

Były to następujące biblioteki:

1. MFRC522.h - *biblioteka odpowiadająca za obsługę breloka RFID.*
2. LiquidCrystal\_I2C.h - *biblioteka odpowiadająca za obsługę wyświetlacza LCD 2x16.*
3. Keypad.h - *biblioteka odpowiadająca za obsługę klawiatury.*
4. Servo.h - *biblioteka odpowiadająca za obsługę servomechanizmu.*
5. SPI.h - *biblioteka odpowiadająca za komunikację z urządzeniami SPI podpiętymi pod Arduino.*
6. OneWire.h - *biblioteka odpowiadająca za komunikację za pomocą protokołu 1-Wire.*
7. DS18B20.h - *biblioteka odpowiadająca za obsługę czujnika DS18B20.*

## 4.2 Komponenty sprzętowe

Wykorzystane platformy sprzętowe, moduły, czujniki, elementy wykonawcze, itp.

Na potrzeby realizacji zadania zostały wybrane, a następnie zamówione w sklepie [botland.com.pl](#) następujące komponenty sprzętowe:

- Moduł Arduino Uno z mikrokontrolerem AVR ATmega328 [1]
- Klawiatura z matrycą 16 x tact switch [6]
- Moduł RFID MF RC522 [8]
- Wyświetlacz LCD 2x16 [10]
- Konwerter I2C dla wyświetlacza LCD [7]
- Serwomechanizm modelarski typu micro [9]
- Cyfrowy czujnik temperatury DS18B20 1-wire [2]
- Dioda LED zielona [5]
- Dioda LED niebieska [4]
- Dioda LED czerwona [3]

Dodatkowo zostały wykorzystane elementy takie jak:

- Płytki stykowe
- Przewody połączeniowe
- Rezystory przewlekane
- Multimetr

## 5 Opis rozwiązania technicznego

### 5.1 Projekt koncepcyjny

Opis zaproponowanego modelu urządzenia, schematu działania aplikacji, funkcjonalności, itp.

Celem projektu było zaprojektowanie, zmontowanie i zaprogramowanie układu zamka elektronicznego z wielopoziomową kontrolą dostępu.

Zamek powinien spełniać kryteria bezpieczeństwa, tak aby mógł zostać wykorzystany w różnych placówkach lub instytucjach, które cenią sobie bezpieczeństwo np. jako zamek drzwi wejściowych, zamek pomieszczenia ograniczonego dostępu, zamek sejfów, skrytki, szafki. Jednak z uwagi na charakterystykę projektu i dobór czujników, system zamka jest dedykowany głównie placówkom medycznych.

System posiada trójstopniową kontrolę dostępu, która jest zastosowana w trybie kaskadowym - użytkownik musi przejść po kolei każdy z etapów blokady by uzyskać dostęp i otworzyć zamek. Informacje o etapie rozbrajania oraz o postęp w ich wyłączeniu, są na bieżąco wyświetlane użytkownikowi.

Dodatkowo podczas próby rozbrojenia danego etapu blokady użytkownik jest informowany o jej powodzeniu lub jego braku, poprzez zapalenie się odpowiednio zielonej lub czerwonej diody LED na okres 3 sekund.

Potencjalny użytkownik, który chce otworzyć zamek, w pierwszej kolejności jest informowany o konieczności zmierzenia temperatury. Powinien on w tym momencie przyłożyć palec dominującej dłoni do powierzchni czujnika i czytać pomiar temperatury z opuszka palca. Program w tym momencie mierzy temperaturę przez 15 sekund. Jeśli w tym czasie temperatura ciała będzie w akceptowalnym zakresie, użytkownik zostanie dopuszczony do następnej blokady. W innym przypadku zostanie poinformowany o nieodpowiedniej temperaturze a program zacznie swoje działanie od początku. W przypadku opuszka palca prawidłowa temperatura wynosi od 25 do 31 stopni Celsjusza.

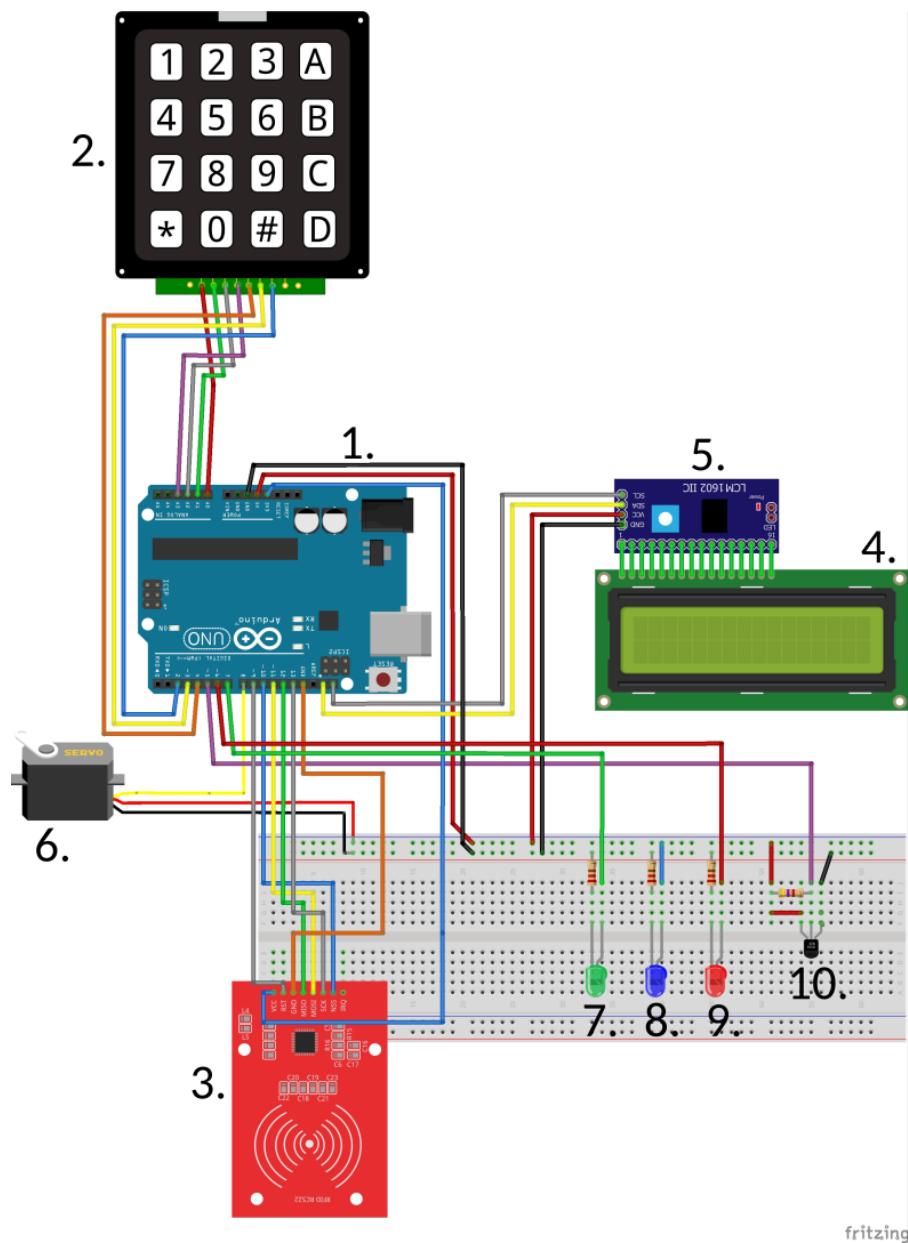
Drugą blokadą jest odczyt poprawnego breloka lub karty magnetycznej. Podobnie jak w przypadku pierwszej blokady, użytkownik zostanie poinformowany o stanie powodzenia operacji. W przypadku niepowodzenia program wraca do początku działania. W przypadku zgodności breloka dostajemy się do ostatniej, trzeciej blokady, którą jest podanie 4 cyfrowego kodu.

Gdy zostanie wpisany poprawny kod użytkownik dostaje odpowiednią informację, oraz komunikat o otwarciu drzwi. Uruchamia się również 5 sekundowy stoper odliczający pozostały czas otwarcia zamka. Serwomechanizm zmienia swoją pozycję o 90 stopni na 5 sekund co symuluje otwarcie się zamka. Po upływie 5 sekund zamek się zamyka a program zaczyna swoją pracę od początku, tak by móc zostać obsługiwany przez następnego użytkownika.

## 5.2 Realizacja techniczna

### 5.2.1 Schemat połączeń

Przed wykonaniem podłączenia zestawu komponentów, dokładnie zaprojektowaliśmy schemat układu połączeń. Został on przedstawiony na rysunku (Rysunek 1).



Rysunek 1: Schemat połączeń środowiska.

### 5.2.2 Opis schematu połączeń

W zaprojektowanym przez nas schemacie podłączenia wszystkich potrzebnych w projekcie modułów, wykorzystaliśmy następujące części, które zostały odpowiednio oznaczone na schemacie (Rysunek 1):

1. Moduł Arduino Uno
2. Klawiatura
3. Moduł RFID MFRC522
4. Wyświetlacz LCD
5. Konwerter I2C dla wyświetlacza LCD
6. Serwomechanizm modelarski typu micro
7. Dioda LED zielona
8. Dioda LED niebieska
9. Dioda LED czerwona
10. Cyfrowy czujnik temperatury DS18B20 1-wire

Podane moduły zostały następnie połączone według zaprojektowanego schematu (Rysunek 1), w następujący sposób:

- Klawiatura została podłączona do kontrolera Arduino z wykorzystaniem pinów 1-7, które obsługują matrycę cyfr 0-9 oraz znaki specjalne "\*" i "#". Ostatni, ósmy pin nie został podłączony, ponieważ symbole "A", "B", "C", "D" nie będą wykorzystywane w projekcie. Schemat podłączenia opisany został w tabeli (Tabela 1):

Klawiatura	Arduino
Pin1	A0
Pin2	A1
Pin3	A2
Pin4	A3
Pin5	4
Pin6	3
Pin7	2

Tabela 1: Schemat połączeń klawiatury

- Moduł RFID RC522 został podłączony do Arduino w sposób przedstawiony w tabeli (Tabela 2):

Moduł RFID	Arduino
RST	9
SDA	10
MOSI	11
MISO	12
SCK	13
GND	GND
3.3V	3.3V

Tabela 2: Schemat podłączeń modułu RFID RC522

- Konwerter I2C dla wyświetlacza LCD został podłączony do Arduino w sposób przedstawiony w tabeli (Tabela 3):

Konwerter I2C	Arduino
SCL	SCL
SDA	SDA
GND	GND
VCC	5V

Tabela 3: Schemat podłączeń konwertera I2C

- Serwomechanizm modelarski typu micro został podłączony do Arduino w sposób przedstawiony w tabeli (Tabela 4):

Serwomechanizm	Arduino
Żółty (sygnał)	D8
Czerwony (+)	5V
Czarny (GND)	GND

Tabela 4: Schemat podłączeń serwomechanizmu modelarskiego



- Diody LED zielona, niebieska, czerwona zostały podłączone w następujący sposób:

Zielona - dłuższą nóżkę diody (anodę) łączymy z wyprowadzeniem nr 7. Krótszą nóżkę (katomę) łączymy przez rezystor z masą (GND)

Czerwona - dłuższą nóżkę diody (anodę) łączymy z wyprowadzeniem nr 6. Krótszą nóżkę (katomę) łączymy przez rezystor z masą (GND)

Niebieska (lampka zasilania) - dłuższą nóżkę diody (anodę) podłączamy do zasilania 5V na płytce stykowej. Krótszą nóżkę (katomę) łączymy przez rezystor z masą (GND)

Diody zostały podłączone do Arduino w sposób przedstawiony w tabeli (Tabela 5):

Diody LED	Arduino
Zielona	D7
Czerwona	D6
Niebieska	5V
Katody(przez rezystor)	GND

Tabela 5: Schemat podłączeń diody LED

- Cyfrowy czujnik temperatury DS18B20 został podłączony do Arduino w sposób przedstawiony w tabeli (Tabela 6) :

Czujnik	Arduino
VDD	5V
GND	GND
DQ	Pin 10

Tabela 6: Schemat podłączeń czujnika temperatury DS18B20

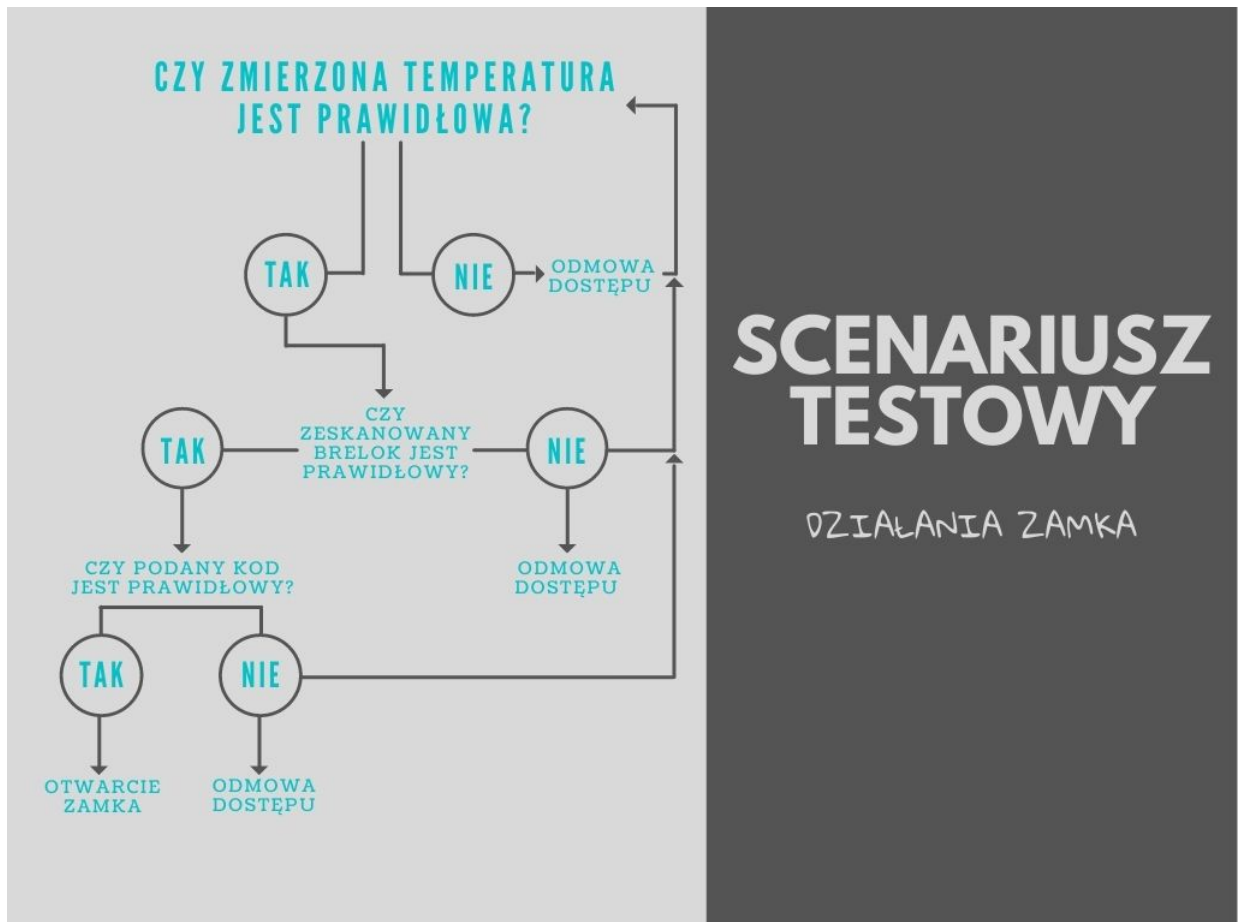
### 5.3 Dokumentacja

Dokumentacja techniczna została opracowana w oddzielnym pliku .pdf zamieszczonym na platformie GitHub ([Link](#)).

Na platformie GitHub zostało również utworzone zbiorcze repozytorium ze wszystkimi plikami opisującymi projekt ([Link](#)).

## 6 Testy techniczne

W oparciu o specyfikację i charakter projektu ułożyliśmy scenariusz testowy sprawdzający poprawne działanie zamka elektronicznego (Rysunek 2). Scenariusz testowy został opracowany w formacie schematu blokowego:



Rysunek 2: Scenariusz testowy w postaci schematu blokowego

### 6.0.1 Opis testu

Potencjalny użytkownik, który chce otworzyć zamek, w pierwszej kolejności jest informowany o konieczności zmierzenia temperatury. Powinien on w tym momencie przyłożyć palec dominującej dłoni do powierzchni czujnika i zczytać pomiar temperatury z opuszka palca. Program w tym momencie mierzy temperaturę przez 15 sekund. Jeśli w tym czasie temperatura ciała będzie w akceptowalnym zakresie, użytkownik zostanie dopuszczony do następnej blokady. W innym przypadku zostanie poinformowany o nieodpowiedniej temperaturze a program zacznie swoje działanie od początku. W przypadku opuszka palca prawidłowa temperatura wynosi od 25 do 31 stopni Celsjusza.

Drugą blokadą jest odczyt poprawnego breloka lub karty magnetycznej. Podobnie jak w przypadku pierwszej blokady, użytkownik zostanie poinformowany o stanie powodzenia operacji. W przypadku niepowodzenia program wraca do początku działania. W przypadku zgodności breloka dostajemy się do ostatniej, trzeciej blokady, którą jest podanie 4 cyfrowego kodu.

Gdy zostanie wpisany poprawny kod użytkownik dostaje odpowiednią informację, oraz komunikat o otwarciu drzwi. Uruchamia się również 5 sekundowy stoper odliczający pozostały czas otwarcia zamka. Serwomechanizm zmienia swoją pozycję o 90 stopni na 5 sekund co symuluje otwarcie się zamka. Po upływie 5 sekund zamek się zamyka a program zaczyna swoją pracę od początku, tak by móc zostać obsłużony przez następnego użytkownika.

Przebieg i wyniki testów oraz działanie zamka można obejrzeć pod [linkiem](#).

## 7 Podsumowanie i wnioski

Projekt był tworzony na przestrzeni dziewięciu tygodni w ramach projektu inżynierskiego. Co tydzień odbywało się spotkanie podsumowujące zrealizowaną pracę.

W ciągu tego okresu udało nam się zrealizować wszystkie założenia i cele projektowe. Zgodnie z ułożonym wcześniej harmonogramem sukcesywnie wypełnialiśmy każdy jego punkt.

Efektem końcowym projektu jest gotowy układ zamka elektronicznego z wielopoziomą kontrolą dostępu. Został on zaprojektowany, skonstruowany oraz zaprogramowany zgodnie ze wszystkimi założeniami wstępnymi.

Układ poprawnie przechodzi wszystkie testy i spełnia swoje główne zadanie jakim jest ograniczenie dostępu nieodpowiednim użytkownikom. Projekt możemy uznać za zakończony sukcesem.

## 8 Załączniki

### Literatura

- [1] Arduino. <https://botland.com.pl/arduino-moduly-glowne/1060-arduino-uno-rev3-a000066-8058333490090.html>.
- [2] Czujnik. <https://botland.com.pl/cyfrowe-czujniki-temperatury/165-czujnik-temperatury-ds18b20-cyfrowy-1-wire-tht.html>.
- [3] Dioda czerwona. <https://botland.com.pl/diody-led/11440-dioda-led-5mm-czerwona-10szt.html>.
- [4] Dioda niebieska. <https://botland.com.pl/diody-led/14490-dioda-led-5mm-niebieska-10szt.html>.
- [5] Dioda zielona. <https://botland.com.pl/diody-led/11144-dioda-led-5mm-zielona-10szt.html>.
- [6] Klawiatura. <https://botland.com.pl/klawiatury-arduino/2895-klawiatura-matryca-16-x-tact-switch.html>.
- [7] Konwerter. <https://botland.com.pl/konwertery-pozostale/2352-konwerter-i2c-dla-wyswietlacza-lcd-hd44780.html>.
- [8] Rfid. <https://botland.com.pl/moduly-i-tag-i-rfid/6765-modul-rfid-mf-rc522-1356mhz-spi-karta-i-b.html>.
- [9] Servo. <https://botland.com.pl/serwomechanizmy/484-serwo-towerpro-sg-90-micro-180-stopni.html>.
- [10] Wyświetlacz. <https://botland.com.pl/wyswietlacze-alfanumeryczne-i-graficzne/223-wyswietlacz-lcd-2x16-znakow-zielony.html>.