Laboratorium z przedmiotu Systemy wbudowane (SW)

Karta projektu – zadanie 7

Nazwa projektu:

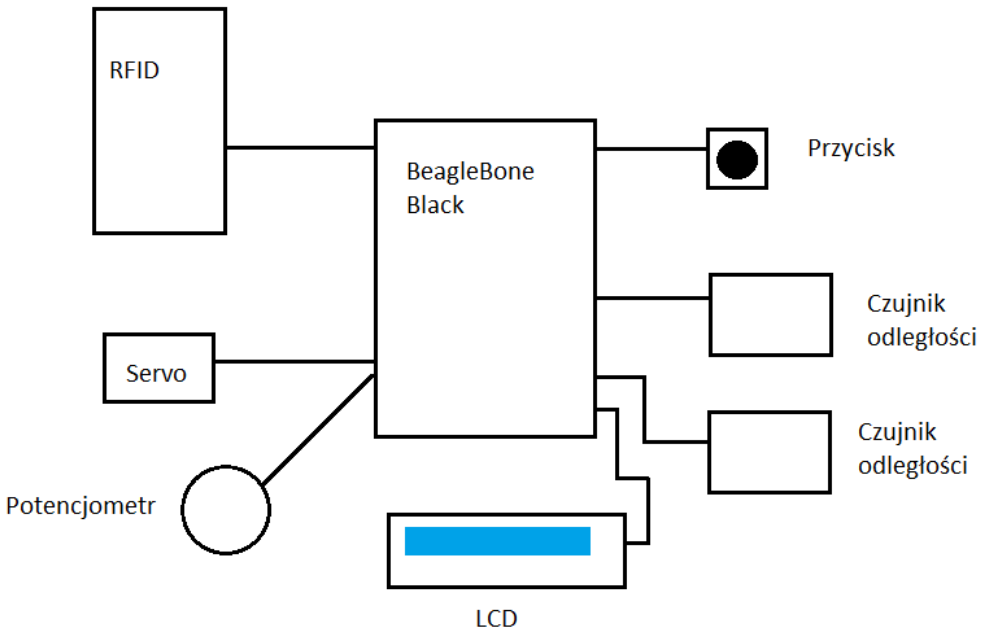
Zamek szyfrowy

Prowadzący:	*Autorzy (tylko nr indeksu):*	*Grupa dziekańska:*	
Antonowicz	136680 136682		I3.2
		Ocena:	

Cel projektu:

Skonstruowanie zamka szyfrowego opartego o identyfikację użytkownika za pomocą RFID oraz zabezpieczenie podwójnym szyfrem wprowadzanym za pomocą potencjometru i czujników odległości. Wykorzystanie bazy danych do przechowywania danych autoryzacji. Opcja administracyjnego trybu dostępu pozwalającego na zmianę danych autoryzacji.

Schemat:



Wykorzystana platforma sprzętowa, czujniki pomiarowe, elementy wykonawcze:

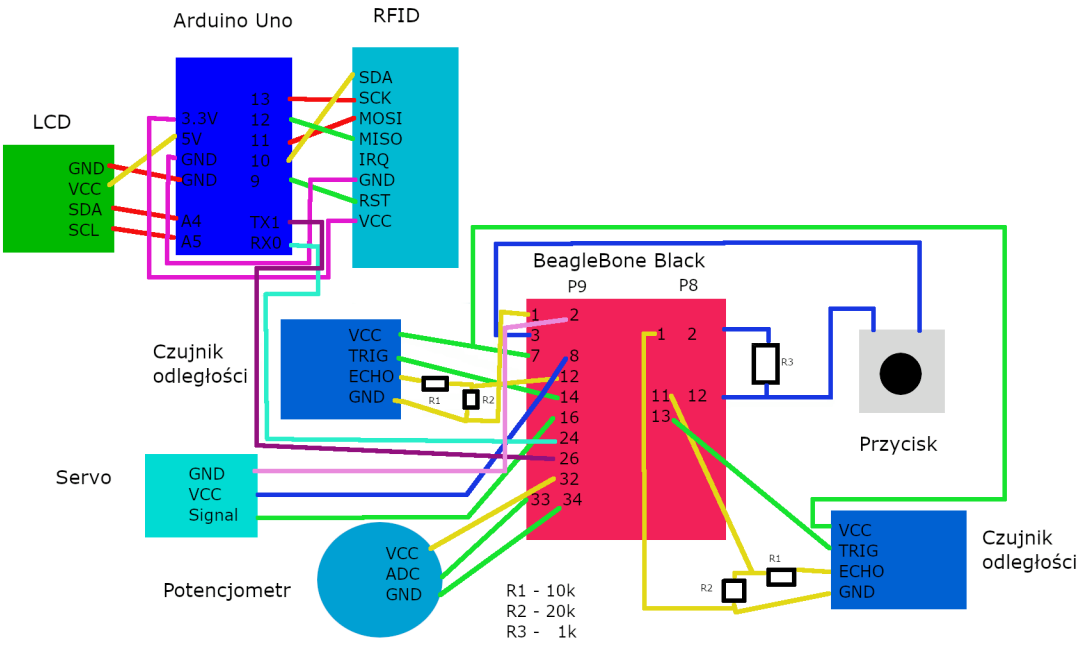
- BeagleBone Black
- potencjometr
- wyświetlacz LCD
- serwomechanizm
- moduł RFID
- czujniki odległości
- przycisk
- Arduino UNO

1 Zakres projektu

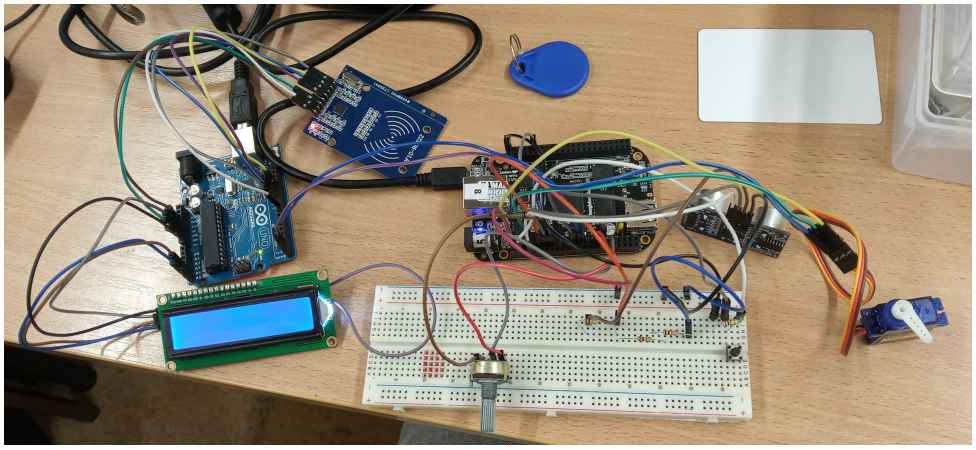
Wykorzystanie modułu RFID do identyfikacji użytkownika, obsługa potencjometru oraz czujników odległości do wprowadzania szyfru, pokazywanie komunikatów na wyświetlaczu LCD, zastosowanie serwomechanizmu jako elementu imitującego zamek, przechowywanie danych autoryzacji w bazie danych.

2 Realizacja

Do realizacji projektu wykorzystano platformy BeagleBone Black i Arduino UNO, do których podłączono czujniki pomiarowe i elementy wykonawcze zgodnie ze schematem na rysunku 1. Moduł RFID MFRC522 oraz wyświetlacz LCD wyposażony w konwerter LCM1602 I2C zostały podłączone do platformy Arduino UNO, natomiast potencjometr, serwomechanizm oraz ultradźwiękowy czujnik odległości HC-SR04 podłączono bezpośrednio do platformy BeagleBone Black, na której uruchomiony jest główny program obsługujący układ.



Rysunek 1: Schemat połączeń w projektowanym układzie



Rysunek 2: Zdjęcie gotowego układu

Początkowo zakładano realizację projektu z wykorzystaniem tylko jednej platformy sprzętowej (BeagleBone Black), jednak podłączenie do niej wyświetlacza LCD oraz modułu RFID okazało się być problematyczne ze względu na ograniczoną dokumentację potrzebnych bibliotek. Problem rozwiązano przy użyciu Arduino UNO – podłączono do niego wyświetlacz i moduł RFID, oraz zapewniono komunikację z BeagleBone Black za pomocą UART. Pozostałe elementy zostały zrealizowane zgodnie z założeniami.

2.1 Interakcja z użytkownikiem

Interakcja systemu z użytkownikiem przebiega następująco:

1. Użytkownik identyfikuje się za pomocą prywatnego transpondera RFID.
2. System weryfikuje użytkownika i wyświetla komunikat z żądaniem wprowadzenia szyfru podstawowego.
3. Użytkownik wprowadza szyfr podstawowy za pomocą układu potencjometru.
4. System sprawdza poprawność szyfru i wyświetla komunikat z żądaniem wprowadzenia szyfru drugorzędnego.
5. Użytkownik wprowadza szyfr drugorzędny za pomocą układu czujników odległości.
6. System sprawdza poprawność szyfru i otwiera zamek.
7. Użytkownik zamyka zamek przez naciśnięcie przycisku.

Możliwy jest również alternatywny scenariusz, w którym odebrany przez czytnik RFID identyfikator należy do administratora systemu:

1. Administrator identyfikuje się za pomocą transpondera RFID.
2. System weryfikuje użytkownika i przełącza się w tryb administracyjny.
3. System odczytuje za pomocą czytnika RFID identyfikator użytkownika, dla którego mają zostać wprowadzone nowe kody dostępu.
4. System wyświetla komunikat z żądaniem utworzenia szyfru podstawowego.
5. Użytkownik wprowadza nowy szyfr podstawowy za pomocą układu potencjometru.
6. System wyświetla komunikat z żądaniem potwierdzenia nowego szyfru.
7. Użytkownik ponownie wprowadza szyfr podstawowy z kroku 5.
8. System sprawdza poprawność szyfru i wyświetla komunikat z żądaniem utworzenia szyfru drugorzędnego.
9. Użytkownik wprowadza nowy szyfr drugorzędny za pomocą układu czujników odległości.
10. System wyświetla komunikat z żądaniem potwierdzenia nowego szyfru.
11. Użytkownik ponownie wprowadza szyfr podstawowy z kroku 9.
12. System sprawdza poprawność szyfru i zapisuje nowe dane uwierzytelniania.
13. System przełącza się w tryb podstawowy.

Zmiana kodów dostępu jest również możliwa bez udziału administratora, jeśli zamek jest w danym momencie otwarty, a transponder użytkownika zostanie ponownie odczytany. W tym przypadku procedura zmiany danych uwierzytelniania przebiega zgodnie z punktami 3–12 scenariusza dostępu administracyjnego.

Komunikaty systemu prezentowane są użytkownikowi za pomocą układu wyświetlacza LCD podłączonego do platformy Arduino UNO, obsługiwanego przy pomocy biblioteki `LiquidCrystal_I2C.h` wykorzystującej protokół I2C do komunikacji z urządzeniem. Tekst do wyświetlenia jest przesyłany z programu głównego, uruchomionego na platformie BeagleBone Black, za pośrednictwem UART.

2.2 Identyfikacja użytkownika

Identyfikacja polega na odczycie przez czytnik RFID numeru UID transpondera użytkownika. Czytnik jest podłączony do platformy Arduino UNO i obsługiwany poprzez protokół SPI za pośrednictwem bibliotek `SPI.h` oraz `MFRC522.h`. Po wykryciu transpondera następuje odczytanie numeru identyfikacyjnego i przesłanie go do programu głównego za pośrednictwem UART. Następnie wykonane zostaje zapytanie do bazy danych, które pozwala stwierdzić, czy użytkownik o danym UID jest zarejestrowany w systemie, a także czy posiada on status administratora.

2.3 Uwierzytelnianie

Projekt zawiera dwa mechanizmy uwierzytelniania użytkownika, uruchamiane kolejno, bezpośrednio po pomyślnej identyfikacji. Pierwszy z nich polega na wprowadzeniu szyfru za pomocą potencjometru, przekręcając jego gałkę na przemian zgodnie i przeciwnie do ruchu wskazówek zegara. Każda zmiana kierunku obrotu, przy pewnej ustalonej dokładności odczytu, powoduje wprowadzenie pojedynczej cyfry szyfru. Kod definiujący proces wprowadzania szyfru podstawowego został przedstawiony na listingu 1.

```
1  def read():
2      # Inicjalizacja
3      [...]
4      passcode = ''
5      # Pętla odczytująca kolejne znaki
6      while len(passcode) < passcodeLength:
7          # Odczyt wartości napięcia z potencjometru
8          v = read_value()
9          previousState = state
10         # Odczytana wartość jest większa od dolnego progu
               dla cyfry większej o 1
11         if state < nSegments - 1 and v > lb[state+1]:
12             if direction < 0:
13                 passcode += hex(state)[2]
14             state += 1
15             direction = 1
16         # Odczytana wartość jest mniejsza od górnego progu
               dla cyfry mniejszej o 1
17         elif state > 0 and v < ub[state-1]:
18             if direction > 0:
19                 passcode += hex(state)[2]
20             state -= 1
21             direction = -1
22     return passcode
```

Listing 1: Fragment kodu sterującego wprowadzaniem szyfru podstawowego

Szyfr drugorzędny jest wprowadzany za pomocą czujników odległości. Kolejne cyfry są generowane na podstawie numeru czujnika oraz wartości odczytanej przez niego odległości. Kod definiujący proces wprowadzania szyfru drugorzędnego został przedstawiony na listingu 2.

```
1  def read():
2      # Inicjalizacja
3      [...]
4      passcode = ''
5      # Pętla odczytująca kolejne znaki
6      while len(passcode) < passcodeLength:
7          # Pętla iterująca po stanach kolejnych czujników
8          for i, state in enumerate(states):
9              # Odczyt odległości z czujnika o numerze i
10             v = read_value(i)
11             # Kwantyzacja wartości odległości
12             newState = states[i] = bisect(bounds, v)
13             if state == nSegments and newState < nSegments:
14                 passcode += hex(i * nSegments + newState)[2]
15     return passcode
```

Listing 2: Fragment kodu sterującego wprowadzaniem szyfru drugorzędnego

Obydwa układy są podłączone do platformy BeagleBone Black i obsługiwane za pomocą bibliotek `Adafruit_BBIO.ADC` oraz `Adafruit_BBIO.GPIO`.

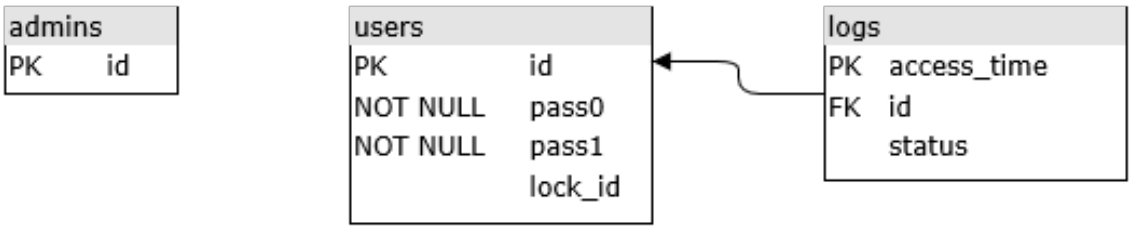
2.4 Sterowanie zamkiem

W zaprojektowanym układzie funkcję zamka pełni serwomechanizm, który jest przełączany pomiędzy stanami odpowiadającymi otwarciu i zamknięciu zamka. Do jego obsługi wykorzystano bibliotekę `Adafruit_BBIO.PWM`. Zgodnie ze scenariuszem opisanym w sekcji 2.1, zamek otwiera się po pomyślnym uwierzytelnieniu użytkownika, natomiast jego zamknięcie następuje po naciśnięciu przycisku. Stan przycisku jest odczytywany przez program za pomocą biblioteki `Adafruit_BBIO.GPIO`.

2.5 Sposób przechowywania danych

Dane wykorzystywane do uwierzytelniania i autoryzacji użytkowników przechowywane są w bazie danych SQLite. Na jej schemat składają się trzy relacje:

- `users`, zawierająca rekordy opisujące użytkowników (identyfikator, wartości funkcji haszujących dla kodów dostępu, przypisany numer zamka),
- `admins`, zawierająca rekordy z identyfikatorami administratorów systemu,
- `logs`, zawierająca rekordy opisujące próby dostępu do systemu (czas dostępu, identyfikator użytkownika, status).



Rysunek 3: Schemat relacyjny bazy danych

3 Wnioski

Celem projektu było skonstruowanie zamka szyfrowego, wykorzystującego kilka różnych czujników do mechanizmu uwierzytelniania użytkownika. Pomimo konieczności zmiany wykorzystywanych urządzeń, cel został osiągnięty. W trakcie realizacji projektu główne trudności były związane z implementacją programu kontrolującego układ, bez możliwości sprawdzania poprawności działania na bieżąco - było to możliwe jedynie podczas zajęć laboratoryjnych. Pomimo prostoty obsługi poszczególnych komponentów z osobna, wyzwaniem okazała się ich wzajemna integracja. W połączeniu z poprzednim problemem objawiało się to błędami w kodzie, których wykrycie i naprawa zajmowała istotną część czasu poświęcanego na projekt. Realizacja pozostałych elementów projektu przebiegła zgodnie z założonym planem.