

Graficzne programowanie mikrokontrolerów

M5Stack / UIFlow

Laboratorium 9

RFID i system kontroli dostępu

Czas trwania: 2 × 45 min**Tryb:** studia stacjonarne**Jednostka:** Katedra Elektrotechniki i Technologii Nadprzewodnikowych**Prowadzący:** dr inż. Michał Łanczont

Cel ćwiczenia

Celem ćwiczenia jest wykorzystanie czytnika RFID Demo Board do identyfikacji kart i budowy prostego systemu kontroli dostępu. Student odczytuje UID, tworzy listę uprawnionych identyfikatorów, realizuje logikę decyzji, sygnalizuje stan systemu na matrycy RGB LED 8×8, steruje wyjściem przekaźnikowym oraz rejestruje zdarzenia.

RFID i UID

Demo Board posiada czytnik RFID oparty na układzie MFRC522. W ćwiczeniu karta lub tag jest używany przede wszystkim jako nośnik identyfikatora UID. Program porównuje odczytany UID z listą identyfikatorów zapisanych w aplikacji.

Obsługę układu **RFID MFRC522** należy realizować za pomocą bloków dostępnych po podpięciu modułu **Unit RFID**

UID nie należy traktować jako bezwarunkowo bezpiecznego mechanizmu uwierzytelniania. Celem laboratorium jest nauka obsługi urządzenia, identyfikacji i logiki decyzji, a nie budowa certyfikowanego systemu bezpieczeństwa.

Matryca RGB LED 8×8 na Demo Board

Na płytce Demo Board znajduje się 64-elementowa matryca RGB LED. Diody są fizycznie rozmieszczone w układzie 8×8, ale elektrycznie tworzą jeden szeregowy łańcuch adresowalnych diod. Oznacza to, że mikrokontroler nie steruje osobnymi liniami wierszy i kolumn. Do całej matrycy doprowadzona jest jedna linia danych, a informacja o kolorze kolejnych diod jest przesyłana szeregowo przez cały łańcuch.

W przykładzie producenta matryca jest sterowana z linii GPIO15, zawiera 64 elementy, a sterowanie jest zgodne z układami klasy WS2811/NeoPixel. W UIFlow należy wykorzystywać funkcje z grupy **Hardware** → **RGB**, a matrycę traktować jako łańcuch 64 kolejno numerowanych diod.

Budowa logiczna matrycy

GPIO15 → LED 0 → LED 1 → LED 2 → ... → LED 63

Numer diody w łańcuchu nie jest tym samym co para współrzędnych (**wiersz**, **kolumna**). Przed tworzeniem znaków graficznych należy sprawdzić rzeczywistą kolejność diod na płytce przez kolejne zapalanie indeksów 0–63. Na tej podstawie można przygotować mapę przejścia z pozycji w macierzy 8×8 na numer diody w łańcuchu.

Zasada sterowania

Dla każdej diody określany jest kolor RGB. Po przygotowaniu wartości dla wybranych elementów należy uaktualnić stan matrycy. W ćwiczeniu nie należy używać maksymalnej jasności wszystkich 64 diod jednocześnie; do sygnalizacji stanu wystarcza niewielka lub średnia jasność.

Przykładowa reprezentacja wzorca 8×8 może być zapisana jako osiem wierszy wartości binarnych, gdzie 1 oznacza diodę świecącą, a 0 – wygaszoną. Kolor jest ustalany oddzielnie dla całego wzorca.

Przykład – czerwony znak X:

```
10000001
01000010
00100100
00011000
00011000
00100100
01000010
10000001
```

Logika dostępu

Typowy algorytm:

karta → UID → wyszukanie → decyzja → GUI/wyjście

Uwaga

Przełącznik w tym ćwiczeniu symuluje rygiel lub lampę sygnalizacyjną. Używaj wyłącznie bezpiecznego niskiego napięcia. Nie podłączaj napięcia sieciowego.

Przebieg ćwiczenia**Algorytm kontroli dostępu**

karta → UID → wyszukanie rekordu → decyzja: dostęp przyznany / odmowa
dostępu → przełącznik + log

Zadanie 1. Odczyt karty

Odczytaj UID dostarczonej karty i wyświetl go na LCD w jednoznacznym formacie, np. jako kolejne bajty szesnastkowe. Sprawdź odczyt dla co najmniej dwóch identyfikatorów.

Sposób realizacji i weryfikacja

1. Zamień każdy bajt UID na dwucyfrowy zapis szesnastkowy i zachowaj stałą kolejność bajtów.
2. Sprawdź kilka kolejnych odczytów tej samej karty - UID powinien być identyczny; różnice oznaczają błąd formatowania lub odczytu.
3. Nie traktuj pustego/brakującego odczytu jako UID złożonego z zer.

Pseudokod / zapis algorytmu:

```
uid_tekst = " ".join(hex2(bajt) dla bajt in uid_bajty)
```

Zadanie 2. Lista uprawnionych kart

Utwórz w programie listę co najmniej trzech rekordów: UID, nazwa użytkownika i poziom dostępu. Po przyłożeniu karty wyświetl nazwę rozpoznanego użytkownika albo komunikat NIEZNANA KARTA.

Sposób realizacji i weryfikacja

1. Przechowuj rekord użytkownika jako zestaw powiązanych danych: UID, nazwa, poziom; unikaj kilku niezależnych list o tej samej długości.
2. Wyszukiwanie kończy się pierwszym dopasowaniem UID; jeśli brak dopasowania, zwróć stan NIEZNANY bez nazwy użytkownika.
3. Dodaj co najmniej dwa poziomy dostępu, aby później decyzja zależała nie tylko od samej obecności UID na liście.

Pseudokod / zapis algorytmu:

```
dla uzytkownik in uzytkownicy:
    if uid == uzytkownik.uid:
        zwroc uzytkownik
zwroc NIEZNANY
```

Zadanie 3. Decyzja: dostęp przyznany / odmowa dostępu

Dla karty spełniającej wymagany poziom dostępu wyświetl DOSTĘP PRZYZNANY, a dla karty nieznanej lub o niewystarczającym poziomie wyświetl ODMOWA DOSTĘPU. Wynik decyzji zapisz w zmiennej stanu. Na tym etapie komunikat jest prezentowany na LCD; w kolejnym zadaniu ten sam stan zostanie odwzorowany na matrycy RGB LED.

Sposób realizacji i weryfikacja

1. Zdefiniuj wymaganą klasę dostępu dla chronionej funkcji i porównuj ją z poziomem użytkownika.
2. Decyzję przechowuj jako zmienną stanu, a dopiero potem aktualizuj GUI i licznik statystyk.
3. Dla NIEZNANY zawsze wybieraj ODMOWA; nie pozostawiaj poprzedniego wyniku na ekranie jako aktualnego.

Pseudokod / zapis algorytmu:

```
if uzytkownik != NIEZNANY and uzytkownik.poziom >= wymagany_poziom:
    wynik = PRZYZNANY
else:
    wynik = ODMOWA
```

Zadanie 4. Sygnalizacja stanu na matrycy RGB LED

Wykorzystaj matrycę 8×8 jako główny sygnalizator stanu terminala kontroli dostępu. Przygotuj co najmniej trzy jednoznacznie różne wzorce graficzne:

- **OCZEKIWANIE NA KARTE** – niebieski wzorec o małej jasności, np. ramka, punkt środkowy lub prosta animacja,
- **DOSTĘP PRZYZNANY** – zielony znak potwierdzenia,
- **ODMOWA DOSTĘPU** – czerwony znak X.

Sposób realizacji i weryfikacja

1. Skonfiguruj sterowanie matrycą jako łańcuchem 64 adresowalnych diod RGB na GPIO15. Ustaw ograniczoną jasność odpowiednią do pracy laboratoryjnej.
2. Wykonaj test numeracji: wygaszaj całą matrycę, a następnie zapalaj kolejno pojedyncze diody o indeksach 0–63. Zapisz kolejność fizycznego położenia diod.
3. Na podstawie testu przygotuj funkcję lub fragment programu, który dla wzorca 8×8 ustawia odpowiednie indeksy LED. Nie zakładaj, że indeksy kolejnych wierszy zawsze rosną od lewej do prawej.
4. Po uruchomieniu programu matryca pokazuje stan OCZEKIWANIE. Po zakończeniu identyfikacji karta powoduje przejście do zielonego lub czerwonego wzorca, zależnie od wyniku decyzji.
5. Po ustalonym czasie sygnalizacji wynikowej wyczyść matrycę i przywróć wzorec OCZEKIWANIE. Poprzedni wynik nie może pozostawać aktywny podczas następnej próby.

Pseudokod / zapis algorytmu:

```
ustaw_jasnosc(20)
pokaz_wzorzec(OCZEKIWANIE, NIEBIESKI)

if wynik == PRZYZNANY:
    pokaz_wzorzec(POTWIERDZENIE, ZIELONY)
else:
    pokaz_wzorzec(KRZYZ, CZERWONY)

po uplywie czasu_sygnalizacji:
    pokaz_wzorzec(OCZEKIWANIE, NIEBIESKI)
```

Wskazówka projektowa

Dobłą praktyką jest oddzielenie danych wzorca od koloru. Ten sam mechanizm rysowania może wówczas wyświetlać różne symbole i barwy bez kopiowania kodu sterującego 64 diodami.

Zadanie 5. Sterowanie przekaźnikiem

Po prawidłowej autoryzacji załącz przekaźnik na krótki, ustalony czas, następnie automatycznie go wyłącz. Przekaźnik nie może pozostać aktywny po restarcie programu ani po nieznanej karcie.

Sposób realizacji i weryfikacja

1. Przekaźnik załączaj dopiero po decyzji PRZYZNANY; nie bezpośrednio w procedurze odczytu UID.
2. Zastosuj timer, po którym wyjście wraca do WYLACZONE bez konieczności kolejnego zdarzenia.
3. W procedurze inicjalizacji i w każdym stanie błędu wymuś WYLACZONE.

Pseudokod / zapis algorytmu:

```
if wynik == PRZYZNANY:
    wlacz_przekaznik()
    odblokuj_do = czas_biezacy + czas_otwarcia
if czas_biezacy >= odblokuj_do:
    wylacz_przekaznik()
```

Zadanie 6. Ochrona przed wielokrotnym odczytem

Jedno przyłożenie karty nie powinno tworzyć wielu zdarzeń. Wprowadź blokadę ponownej obsługi tego samego UID do chwili odsunięcia karty lub upływu ustalonego czasu.

Sposób realizacji i weryfikacja

1. Po obsłudze UID ustaw stan KARTA_OBECNA; kolejny odczyt tego samego UID nie tworzy nowego zdarzenia.
2. Powrót do OCZEKIWANIE_KARTY następuje po wykryciu braku karty przez określony czas lub po ustalonym przekroczeniu czasu oczekiwania.
3. Sprawdź długie przytrzymanie karty oraz szybkie odsunięcie i ponowne przyłożenie.

Pseudokod / zapis algorytmu:

```
OCZEKIWANIE_KARTY -> odczytaj UID -> PRZETWARZANIE -> KARTA_OBECNA
KARTA_OBECNA -> karta_usunieta -> OCZEKIWANIE_KARTY
```

Zadanie 7. Rejestr zdarzeń

Przechowuj licznik udanych i odrzuconych prób. Na ekranie dodaj stronę statystyk: ostatni UID, użytkownik, wynik, liczba PRZYZNANY i liczba ODMOWA.

Sposób realizacji i weryfikacja

1. Dla każdego zdarzenia zapisz co najmniej UID, rozpoznaną nazwę/NIEZNANY, wynik i numer kolejny lub czas od uruchomienia.
2. Liczniki PRZYZNANY i ODMOWA zwiększaj raz na obsłużone zdarzenie, po mechanizmie anti-powtórzeniowym.
3. Ekran statystyk nie może sterować przełącznikiem; jest wyłącznie prezentacją zgromadzonych danych.

Pseudokod / zapis algorytmu:

```
licznik_zdarzen += 1
if wynik == PRZYZNANY: przyznane += 1
else: odmowy += 1
```

Wymagany rezultat

System rozpoznaje uprawnione identyfikatory, odrzuca pozostałe, jednoznacznie sygnalizuje stan na matrycy RGB LED, steruje wyjściem tylko po poprawnej decyzji i nie zlicza wielokrotnie jednego przyłożenia karty.

Zadanie końcowe

Integracja zadań etapowych

Zadanie końcowe jest bezpośrednim rozwinięciem zadań 1–7. Odczyt UID z zadania 1 zasila listę użytkowników z zadania 2, decyzja dostępu z zadania 3 wybiera wzorzec matrycy RGB z zadania 4 i warunkuje sterowanie przełącznikiem z zadania 5. Automat z zadania 6 zapobiega wielokrotnemu zaliczaniu tej samej karty, a rejestr z zadania 7 gromadzi wyniki wszystkich prób.

Zbuduj terminal kontroli dostępu z co najmniej trzema użytkownikami. Każdy rekord zawiera UID, nazwę i poziom dostępu. Wybrany przełącznik może zostać aktywowany tylko dla osoby o

odpowiednim poziomie. Na ekranie używaj polskich komunikatów, np. **DOSTĘP PRZYZNANY**, **ODMOWA DOSTĘPU**, **NIEZNANA KARTA**. Matryca RGB LED ma równolegle sygnalizować stan terminala: oczekiwanie na kartę, dostęp przyznany oraz odmowę dostępu za pomocą różnych wzorców i kolorów. Po zakończeniu sygnalizacji wyniku system automatycznie wraca do stanu oczekiwania. Dodaj licznik wszystkich prób, licznik prób zaakceptowanych i odrzuconych oraz ekran ostatniego zdarzenia.

Rozszerzenie: umożliw wyzerowanie statystyk wyłącznie po użyciu karty o poziomie administratora i dodatkowym potwierdzeniu przyciskiem.

Kryteria zaliczenia

- stabilny odczyt UID,
- lista co najmniej trzech rekordów,
- decyzja: dostęp przyznany / odmowa dostępu,
- poprawna konfiguracja i test numeracji 64 diod matrycy RGB,
- czytelna sygnalizacja stanów OCZEKIWANIE, DOSTĘP PRZYZNANY i ODMOWA DOSTĘPU na matrycy 8×8,
- automatyczny powrót matrycy do stanu oczekiwania,
- bezpieczne sterowanie przekaźnikiem,
- eliminacja wielokrotnego odczytu,
- rejestr i prezentacja statystyk.

Sprawozdanie

Podstawą zaliczenia ćwiczenia jest sprawozdanie dotyczące wykonania zadania końcowego. Zadania etapowe są zaliczane na zasadzie demonstracji działania programu na stanowisku. Sprawozdanie należy przesłać w formie pliku *.pdf oraz pliku projektu programu do weryfikacji. W dokumencie należy przedstawić:

- kod programu w UIFlow,
- zdjęcia działającego stanowiska lub czytelne zrzuty ekranu,
- krótki opis zastosowanego algorytmu wraz ze schematem blokowym, diagramem stanów lub pseudokodem,
- opis sposobu testowania programu,
- wnioski.