

Graficzne programowanie mikrokontrolerów

M5Stack / UIFlow

Laboratorium 3

GPIO – wejścia i wyjścia cyfrowe

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

Cel ćwiczenia

Celem ćwiczenia jest poznanie zasad obsługi wejść i wyjść cyfrowych w systemie M5Stack. Głównym elementem wejściowym jest enkoder inkrementalny z wbudowanym przyciskiem dostępny na płytce **DEMO BOARD**. Student odczytuje stany logiczne, wykrywa zbocza, eliminuje drgania styków, rozpoznaje kierunek obrotu na podstawie sygnałów kwadraturowych A/B oraz wykorzystuje enkoder i jego przycisk do sterowania wyjściami cyfrowymi.

Sygnał cyfrowy

Wejście cyfrowe rozróżnia dwa poziomy logiczne. W programie są one zwykle interpretowane jako 0/1, False/True albo LOW/HIGH. Wyjście cyfrowe pozwala ustawić odpowiedni poziom na pinie lub sterować elementem pośredniczącym, np. przekaźnikiem.

Stan wejścia nie powinien pozostawać nieokreślony. W praktyce stosuje się rezystor *pull-up* lub *pull-down*. W przypadku elementów mechanicznych trzeba dodatkowo uwzględnić drgania styków, przez które jedno fizyczne naciśnięcie przycisku lub pojedyncza zmiana stanu enkodera może zostać zinterpretowana wielokrotnie.

W *UIFlow* wejścia i wyjścia cyfrowe można obsługiwać za pomocą obiektów sprzętowych odpowiadających pinom GPIO. Przed użyciem należy skonfigurować odpowiedni pin jako wejście albo wyjście oraz, jeżeli jest to potrzebne, włączyć podciąganie. Numery GPIO wykorzystywane na stanowisku należy przyjąć zgodnie ze schematem połączeń przygotowanym dla laboratorium.

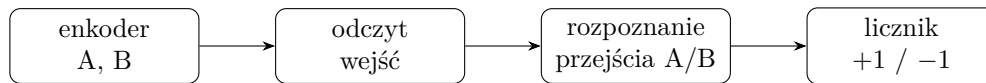
Enkoder inkrementalny z przyciskiem

Enkoder na płytce **DEMO BOARD** jest elementem cyfrowym. Udostępnia trzy niezależne sygnały: dwa kanały enkodera **A** i **B** oraz sygnał **BUTTON** pochodzący od przycisku umieszczonego w osi pokrętła. Na złączu enkodera płytki są one opisane jako **IN_A**, **IN_B**, **BUTTON** oraz **GND**.

Połączenie stanowiska

Kanałów A i B nie należy traktować jako wejść analogowych. Do ćwiczenia połącz **IN_A**, **IN_B** i **BUTTON** z trzema wejściami cyfrowymi wskazanymi w schemacie stanowiska oraz połącz wspólną masę. Przed uruchomieniem programu sprawdź, czy wybrane wejścia mogą pracować z wymagany podciąganiem.

Enkoder inkrementalny nie podaje bezpośrednio bezwzględnego położenia pokrętła. Podczas obrotu generuje dwa przebiegi cyfrowe przesunięte w fazie. Na podstawie kolejności zmian A i B można określić kierunek obrotu oraz zliczać kolejne kroki.



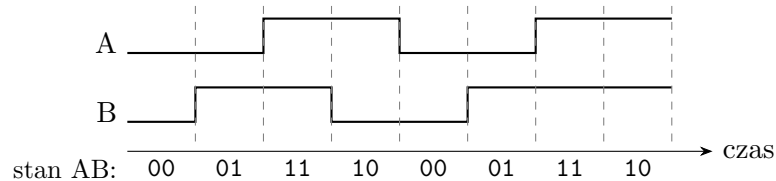
Dla jednego kierunku obrotu poprawna sekwencja stanów może mieć postać:

00 → 01 → 11 → 10 → 00

a dla kierunku przeciwnego:

00 → 10 → 11 → 01 → 00.

To, która sekwencja oznacza obrót w prawo, zależy od sposobu podłączenia kanałów i przyjętej konwencji. Po uruchomieniu programu należy to sprawdzić doświadczalnie.



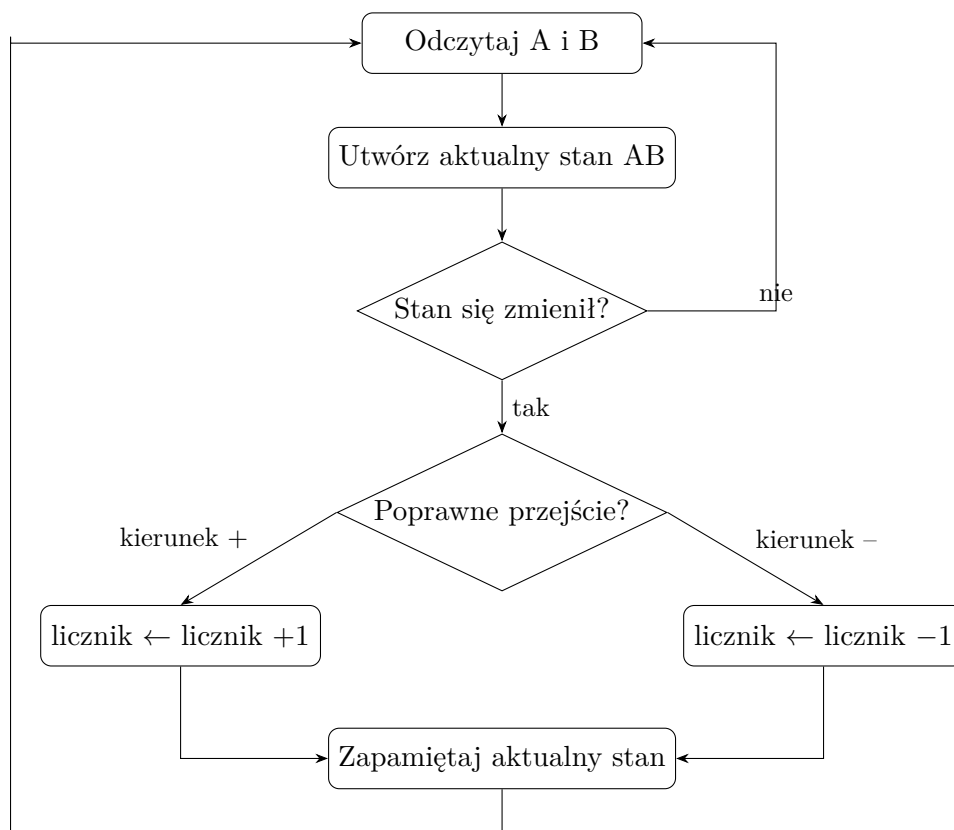
Co oznacza jeden krok?

W jednym pełnym cyklu kwadraturowym występują cztery poprawne przejścia pomiędzy stanami A/B. Program może zliczać tylko jedno wybrane zbocze, dwa zbocza albo wszystkie cztery przejścia. W ćwiczeniu należy przyjąć jedną konsekwentną definicję kroku, tak aby jeden wyczuwalny skok pokrętła powodował przewidywalną zmianę wartości użytkowej.

Najprostsza metoda programowa polega na zapamiętaniu poprzedniego stanu A/B i sprawdzaniu, czy kolejne przejście należy do sekwencji dla kierunku dodatniego lub ujemnego.

Stan poprzedni	Stan aktualny	Zmiana licznika
00	01	+1
01	11	+1
11	10	+1
10	00	+1
00	10	-1
10	11	-1
11	01	-1
01	00	-1

Pozostałe przejścia, np. 00 → 11, oznaczają pominięcie stanu albo zakłócenie i w prostym algorytmie można je zignorować.



Pseudokod dekodowania:

```

poprzedni_stan = odczytaj_AB()
licznik = 0

petla:
    biezacy_stan = odczytaj_AB()
    if biezacy_stan != poprzedni_stan:
        if przejście_w_przód(poprzedni_stan, biezacy_stan):
            licznik = licznik + 1
        else if przejście_w_tył(poprzedni_stan, biezacy_stan):
            licznik = licznik - 1
        poprzedni_stan = biezacy_stan
  
```

Przycisk enkodera jest niezależnym wejściem cyfrowym. Powinien być obsługiwany z detekcją zbocza i eliminacją drgań styków, dzięki czemu pojedyncze naciśnięcie wywołuje dokładnie jedno zdarzenie.

Przebieg ćwiczenia

Tor przetwarzania sygnałów cyfrowych

A/B + przycisk → detekcja zmian → dekodowanie kierunku → licznik / wybór → wyjścia + GUI

Zadanie 1. Odczyt przycisku enkodera

Odczytaj stan przycisku zintegrowanego z enkoderem i wyświetl na ekranie zarówno wartość logiczną wejścia, jak i jej interpretację: ZWOLNIONY albo WCIŚNIĘTY. Ustal doświadczalnie, czy przycisk jest aktywny stanem wysokim, czy niskim.

Sposób realizacji i weryfikacja

1. Odczytaj stan wejścia przed naciśnięciem i podczas naciśnięcia.
2. Nie zakładaj z góry logiki aktywnej HIGH lub LOW – określ ją na podstawie pomiaru.
3. Sprawdź działanie dla krótkiego naciśnięcia i dłuższego przytrzymania.

Zadanie 2. Detekcja zbocza i eliminacja drgań styków

Zmień program tak, aby zmienna `licznik_nacisniec` zwiększała się dokładnie raz przy każdym naciśnięciu przycisku, niezależnie od czasu jego przytrzymania.

```
biezacy = odczytaj_przycisk()
if biezacy == AKTYWNY and poprzedni == NIEAKTYWNY:
    czekaj_ms(czas_eliminacji_drgan)
    if odczytaj_przycisk() == AKTYWNY:
        licznik_nacisniec += 1
poprzedni = biezacy
```

Weryfikacja

Wykonaj co najmniej dziesięć naciśnień o różnym czasie trwania. Licznik ma zwiększyć się dokładnie o liczbę rzeczywistych naciśnień.

Zadanie 3. Obserwacja stanów A/B enkodera

Odczytuj jednocześnie kanały A i B. Wyświetlaj je na ekranie jako dwubitowy stan AB. Powoli obracaj enkoder w obu kierunkach i zanotuj kolejność pojawiających się stanów.

Sposób realizacji i weryfikacja

1. Wyświetlaj osobno stan A, stan B oraz połączony zapis 00, 01, 10 lub 11.
2. Przy bardzo wolnym obrocie sprawdź, czy obserwowana kolejność odpowiada jednej z sekwencji opisanych we wstępie.
3. Powtórz próbę dla przeciwnego kierunku i potwierdź odwrócenie kolejności stanów.

Zadanie 4. Zliczanie obrotu i rozpoznawanie kierunku

Zaimplementuj licznik położenia względnego enkodera. Obrót w prawo powinien zwiększać wartość, a obrót w lewo ją zmniejszać. Dozwolone są dwie metody: odczyt kanału B na wybranym zboczu kanału A albo analiza kolejnych stanów A/B na podstawie tabeli przejść.

Sposób realizacji i weryfikacja

1. Przyjmij i zapisz, co w Twoim programie oznacza jeden krok użytkowy.
2. Ogranicz wartość licznika do zakresu, np. 0...100.
3. Jeżeli obrót w prawo zmniejsza licznik, odwróć interpretację kierunku – nie zmieniaj przypadkowo całego algorytmu.
4. Sprawdź szybki obrót w obu kierunkach i zwróć uwagę na ewentualne pominięte lub błędne przejścia.

Zadanie 5. Enkoder jako element nastawczy

Wykorzystaj licznik z poprzedniego zadania jako wartość `nastawa` w zakresie 0...100. Obrót zmienia nastawę, natomiast krótkie naciśnięcie przycisku ustawia wartość domyślną 50. Na ekranie przedstaw nastawę liczbowo i w postaci paska.

```
if krok == +1: nastawa += 1
if krok == -1: nastawa -= 1
nastawa = ogranicz(nastawa, 0, 100)

if wykryto_nacisniecie:
    nastawa = 50
```

Zadanie 6. Sterowanie wyjściami za pomocą enkodera

Wykorzystaj enkoder jako prosty interfejs operatora do sterowania trzema wyjściami przełącznikowymi. Obrót wybiera element R1, R2 lub R3, a krótkie naciśnięcie przycisku przełącza stan aktualnie wybranego wyjścia. Po uruchomieniu programu wszystkie przełączniki muszą być wyłączone.

Sposób realizacji i weryfikacja

1. Zmienna **wybrane_wyjscie** przyjmuje wartości 1–3 i jest zmieniana obrotem enkodera.
2. Naciśnięcie przycisku zmienia wyłącznie stan aktualnie wskazanego przełącznika.
3. Ekran prezentuje numer wybranego wyjścia i rzeczywiste stany R1–R3.
4. Długie przytrzymanie przycisku, np. ponad 1,5 s, wykorzystaj jako polecenie **WYŁĄCZ WSZYSTKO**.
5. Do prób używaj wyłącznie bezpiecznych obwodów SELV wskazanych przez prowadzącego; nie podłączaj napięcia sieciowego.

Wymagany rezultat

Program poprawnie rozpoznaje naciśnięcie przycisku, kierunek obrotu enkodera i jego kolejne kroki. Enkoder może być używany jako element nastawczy i jako interfejs wyboru wyjścia, a po restarcie wszystkie wyjścia pozostają w stanie bezpiecznym.

Zadanie końcowe

Integracja zadań etapowych

Zadanie końcowe ma być bezpośrednim rozwinięciem programów z zadań 1–6. Obsługa przycisku wykorzystuje detekcję zbocza i eliminację drgań, obrót enkodera korzysta z jednego wspólnego mechanizmu dekodowania A/B, a stany przełączników są przechowywane w zmiennych programu i prezentowane na ekranie.

Zbuduj **panel sterowania trzema wyjściami**. Obrót enkodera wybiera R1, R2 lub R3. Krótkie naciśnięcie przełącza stan wybranego wyjścia. Długie przytrzymanie przycisku realizuje polecenie **WYŁĄCZ WSZYSTKO**. Na ekranie pokaż bieżące stany A/B, wartość licznika enkodera, wybrane wyjście oraz stany R1–R3. Program po każdym uruchomieniu musi jawnie ustawić wszystkie wyjścia w stan wyłączony.

Rozszerzenie: dodaj drugi tryb pracy, w którym enkoder nie wybiera przełącznika, lecz ustawia wartość **nastawa** 0...100; krótkie naciśnięcie przełącza pomiędzy trybem **WYBÓR WYJŚCIA** i **NASTAWA**.

Kryteria zaliczenia

- poprawny odczyt przycisku enkodera i identyfikacja aktywnego poziomu logicznego,
- detekcja pojedynczego naciśnięcia z eliminacją drgań styków,
- poprawny odczyt kanałów A i B oraz rozpoznawanie kierunku obrotu,

- konsekwentne zliczanie kroków enkodera,
- wykorzystanie enkodera i przycisku jako interfejsu operatora,
- bezpieczne sterowanie co najmniej trzema wyjściami cyfrowymi.

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,
- opis sposobu odczytu i dekodowania sygnałów A/B,
- schemat blokowy, diagram stanów lub pseudokod zastosowanego algorytmu,
- opis sposobu testowania programu, w tym test obrotu w obu kierunkach i test przycisku,
- wnioski.