

Graficzne Programowanie Mikrokontrolerów

Ćwiczenie 2

Enkoder przyrostowy

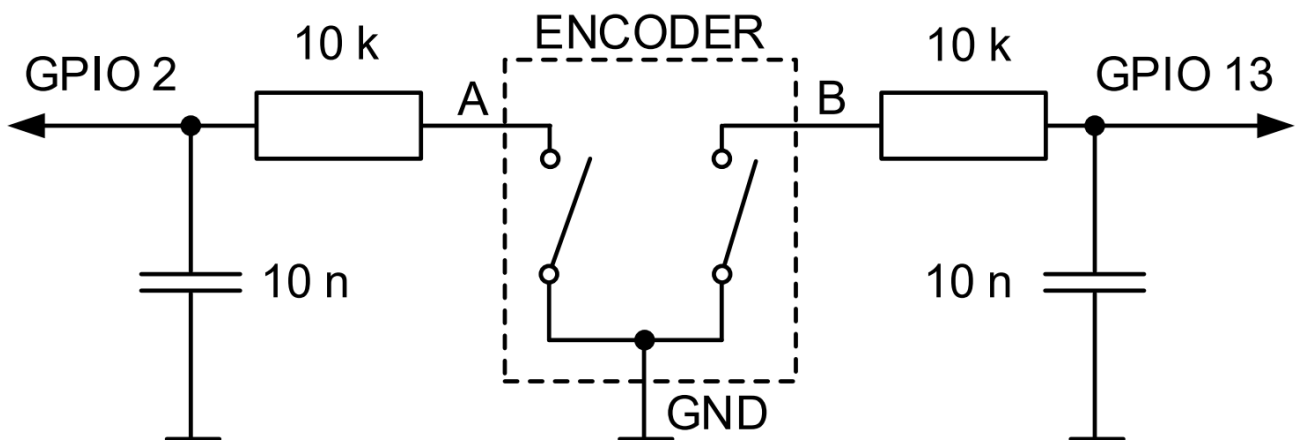
Wstęp

Z racji przyjętej przez M5Stack modelu biznesowego bloki programowe dedykowane są dla oferowanych przez firmę produktów. Jeżeli za pomocą mikrokontrolerów M5Stack ma być oprogramowany produkt firm trzecich to może być konieczne zapisanie własnego kodu obsługi elementu. W wielu przypadkach jednak o ile element oparty jest na tych samych układach elektronicznych co odpowiednik M5Stack jest możliwe że dostarczone bloki pozwolą na obsługę elementu.

Dobrym przykładem konieczności opracowania własnego kodu obsługi jest enkoder przyrostowy. Ćwiczenie oparte zostało na:

- moduł developerski M5Core
- zestaw edukacyjny Demo Board
- dwa rezystory - około 10 k Ω , 0.125 W
- dwa kondensatory - 10 nF, 63 V
- sześć przewodów łączeniowych (goldpin)

Ze względu na zasadę działania enkodera przyrostowego oraz jego jakość konieczne staje się zastosowanie filtra przeciwzakłóceńowego w celu ograniczenia wpływu zakłóceń komutacyjnych enkodera na odczyt stanów logicznych portów.



Rys. 1: Schemat połączeń filtra przeciwzakłóceńowego z enkoderem i portami mikrokontrolera

Kanały enkodera należy podłączyć do portów 2 i 13, a przycisk do portu 15 mikrokontrolera.

1 Zadanie pierwsze



Rys. 2: Interfejs programu

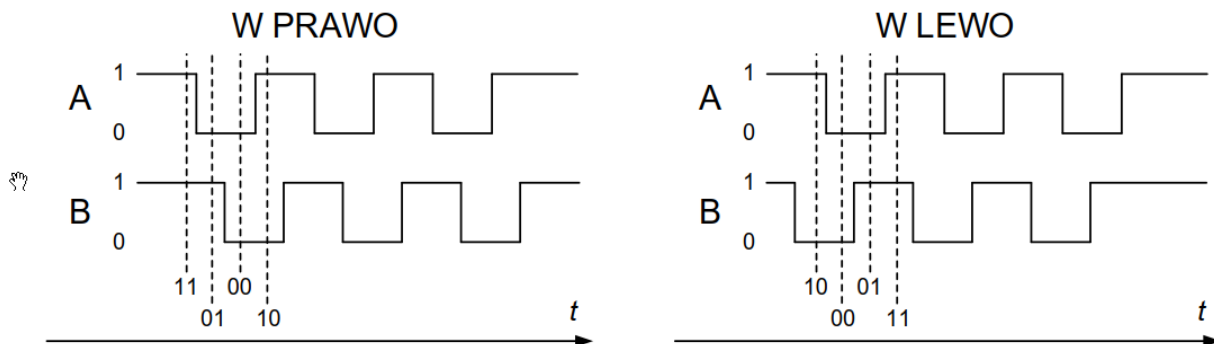
Zakres ćwiczenia obejmuje:

- Należy sprawdzić, czy firmware mikrokontrolera Core jest aktualny. Wykonać aktualizację, jeśli pojawiła się nowa wersja.
- Napisać i uruchomić program pozwalający obserwować na ekranie LCD mikrokontrolera sygnały cyfrowe na wejściach GPIO 2, 13, 15.
- GPIO 2, 13, i 15 należy skonfigurować (blok Hardware/PIN/Init) jako wejścia cyfrowe (mode: IN) z rezystorami ustalającymi domyślny stan wysoki (pull: PULL-UP).
- Przeanalizować kolejność zmian wartości sygnałów A i B przy dowolnym obracaniu enkodera w lewo i w prawo.

2 Zadanie drugie

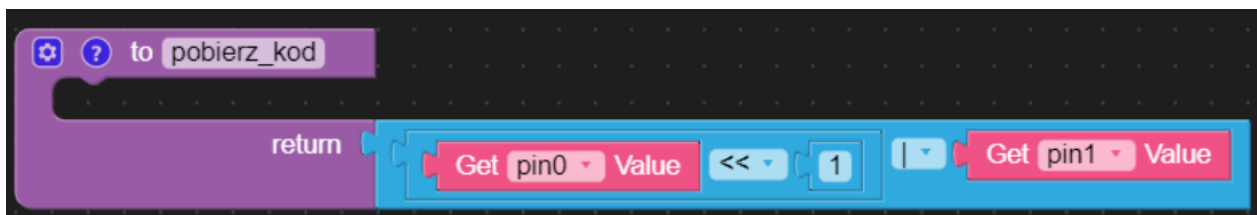
Napisać, uruchomić i przetestować program pozwalający nastawiać wartość liczbową całkowitą w przedziale $\langle -100, 100 \rangle$ za pomocą pokrętła enkodera. Przycisk pokrętła enkodera wykorzystać jako klawisz uruchamiający procedurę nastawiania/zatwierdzający nastawioną wartość.

Działanie enkodera: podczas obracania pokrętła enkodera w prawo (zgodnie z ruchem wskazówek zegara = CW) generowane są sygnały pokazane na rysunku poniżej. Podczas obracania pokrętła enkodera w lewo (przeciwnie do ruchu wskazówek zegara = CCW) zmienia się kolejność zamykania styków i generowane są sygnały o zmienionej kolejności.



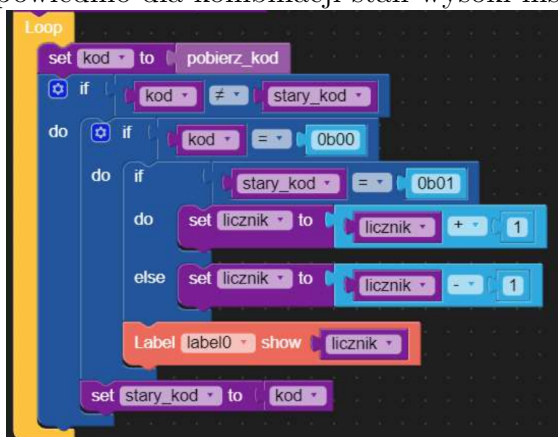
Rys. 3: Przebiegi czasowe sygnałów na portach enkodera przy obrocie zgodnym i przeciwnym

W najprostszej realizacji rozpoznanie kierunku polega więc na ustaleniu, która z sekwencji kodów została odebrana (np. który kod poprzedził kod 00, jeśli 01 to kierunek w prawo, jeśli 10 to w lewo). Bardziej zaawansowane algorytmy sprawdzają poprawność całej sekwencji w celu wyeliminowania błędnych sygnałów związanych z drganiami styków).



Rys. 4: Propozycja funkcji do odczytania wartości z pinów A i B enkodera i obliczenia kodu

Pin0 (GPIO 2) podłączony jest do wyjścia A enkodera, pin1 (GPIO 13) do wyjścia B. Wartość kodu jest równa 3 (dwójkowo 0b11), jeśli odebrano dwa stany wysokie. Odpowiada to pozycji spoczynkowej enkodera. Wartość kodu 0 (0b00) wystąpi dla dwóch stanów niskich. Kody 2 (0b10) i 1 (0b01) pojawiają się odpowiednio dla kombinacji stan wysoki-niski i niski-wysoki.



1. jest jakaś zmiana "kod"
2. jeżeli po kod = 0b00 pojawił się po 0b01 to jest kierunek w prawo
3. w przeciwnym razie jest to kierunek w lewo

3 Zadanie trzecie

- Należy przenieść program zrealizowany na podstawie instrukcji nr 1 (zadanie w rozdziale 6) na mikrokontroler typu Core. Operacja taka nosi nazwę konwersji lub portu oprogramowania.
- W kolejnym kroku trzeba gruntownie sprawdzić poprawność działania skonwertowanego programu.
- Następnie zmodyfikować program tak, aby nastawianie wartości początkowej czasu (HH:MM:SS) odbywało się za pomocą enkodera.