



## KATEGORIA WAGO

## ETAP ELIMINACJI

Projekty muszą zawierać w nazwie indywidualny **KOD ZAWODNIKA** i zostać przesłane do **18:20** na mail: [mistrzostwaplc@pwr.edu.pl](mailto:mistrzostwaplc@pwr.edu.pl). Prace zawierające dane osobowe nie będą sprawdzane przez Jury. Zmienne wykorzystane w opisie programów są już wprowadzone do szablonu w odpowiednich funkcjach wykonywalnych.

Treść zadania	
Przygotować program w szablonie projektu <b>e!COCKPIT</b> realizujący poniższe wymagania.	
1	Napisać program, w którym dwukrotne wykrycie zbocza narastającego na wejściu <b>isTrig</b> w czasie 2 sekund powoduje trzykrotne włączenie się zmiennej <b>PulseLamp</b> , gdzie w stanie wysokim trwa 1s z przerwą między pulsami wynoszącą 1s. Gdy włączone jest wejście <b>isOpen</b> długość przerwy między pulsami zwiększa się o 2s. Dodatkowo jeśli wyłączone jest wejście <b>isClose</b> długość pulsu zwiększana jest do 2s. Cały układ działa, jedynie jeśli na wejściu zmiennej <b>isWorking</b> występuje stan wysoki.
2	Napisać program, w którym za pomocą jednego wejścia <b>isTrig</b> możemy sterować trzema wyjściami <b>Lamp</b> , <b>Engine</b> , <b>Fan</b> . Wybór urządzenia do przełączenia zależy od tego ile razy został wciśnięty przycisk <b>isTrig</b> (zliczanie zbocz narastających). Wybranie danego urządzenia skutkuje przełączeniem w zależności od aktualnego stanu powoduje jego zmianę z 0 na 1 lub z 1 na 0. Zliczanie wciśnień kończy się po upływie ponad T#1000ms od ostatniego wciśnięcia przycisku.  <b>Lamp</b> - przełączenie przy 1 wciśnięciu; <b>Engine</b> - przełączenie przy 2 wciśnięciach; <b>Fan</b> - przełączenie przy 3 wciśnięciach; brak działania - 0 lub 4+ wciśnień
3	Napisać program, w którym za pomocą zbocza narastającego wejścia <b>isOn</b> uruchamiana jest sekwencja zapalanych po sobie 8 diod <b>Diodes ([0..7] array of BOOL)</b> . Zapalają się one w odstępie sekundowym i każda z nich podtrzymuje stan wysoki na okres 1 sekund. Po zapaleniu się ostatniej diody mają one zapalać się w odwrotnej kolejności, aż do początku. Po wykonaniu podanej sekwencji program czeka na ponowne uruchomienie przy pomocy wejścia <b>isOn</b> . Dodatkowo, jeśli na wejściu <b>isOn</b> wykryte zostanie zbocze narastające w trakcie wykonywania sekwencji program przerywa jego działanie i zaczyna wykonywanie sekwencji od początku.
4	Napisać program, który będzie realizować działanie sterowania PWM. Po 2 sekundach od pojawienia się stanu wysokiego na wejściu <b>isWorking</b> rozpoczyna się działanie układu PWM. Sterowanie odbywa się z częstotliwością zależną od wartości analogowej <b>aFreq</b> oraz wypełnieniem od wartości analogowej <b>aFill</b> . Modulowany sygnał z funkcji PWM podać na wyjścia <b>PWMdiodeOn</b> oraz <b>PWMdiodeOff</b> , w taki sposób, że stan wysoki z zaprogramowanej funkcji pojawia się na zmiennej <b>PWMdiodeOn</b> , a negowany sygnał na <b>PWMdiodeOff</b> . Program ma działać, tylko jeżeli wejście <b>isWorking</b> ma stan wysoki.  Dodatkowo: Wartość zmiennej analogowej <b>aFreq</b> w zakresie (0:1000) odpowiada częstotliwości (1:10) Hz. Wartość zmiennej analogowej <b>aFill</b> w zakresie (0:1000) odpowiada wypełnieniu (100:1000)ms.



5

Program zlicza liczbę zboczy narastających na zmiennej **Triggered**. W zależności od ich liczby wykonywana jest jedna z podanych operacji:

0 – nic,

1 – wyświetlanie wartości **Equation** w postaci bitowej na diodach **Diodes** ([0..7] array of BOOL) przez 4 sekundy,

2 – dodanie wartości 2 do zmiennej **Equation**,

3 – odjęcie wartości 3 od zmiennej **Equation**,

4 – wyzerowanie zliczanych zboczy (wykonywane automatycznie)

Wybraną operację wykonuje się raz to zatwierdzeniu wyboru zmienną **isConfirmed**. Wyświetlanie polega na przekonwertowaniu wartości **Equation** do postaci bitowej z zapewnieniem, że pierwsza dioda odpowiada za znak liczby (0 – liczba dodatnia, 1 – liczba ujemna). Zmienna **Equation** ma ograniczony zakres ze względu na możliwość jej zapisu na diodach. Kodowanie jest zgodne z kodem uzupełnień zapisu do dwóch w zakresie **(-16:15)**. W przypadku przepełnienia (przekroczenie dolnego, bądź górnego zakresu zmiennej w postaci bitowej) należy dokonać tzw. zapętlenia, czyli zapisu nadmiarowej części liczby jako wartość mieszcząca się w zakresie. Np. po dodaniu 2 do **Equation** = 14 wynikiem będzie **Equation** = -16 lub odjęcie 3 od **Equation** = -15 wynikiem będzie **Equation** = 14.

Przykład zapalenia diod:

00000 (diody od 0 do 4) – wartość **Equation** = 0

10000 – wartość **Equation** = -16

01111 – wartość **Equation** = 15

00001 – wartość **Equation** = 1