

INFORMATYCZNE SYSTEMY STEROWANIA

INSTRUKCJA ĆWICZENIA LABORATORYJNEGO

Ćwiczenie nr 9

Wonderware InTouch – zmienne, połączenia animacyjne, skrypty

Czas trwania: 2h

I. CHARAKTERYSTYKA ĆWICZENIA

Cel i zakres:

Ćwiczenie jest kontynuacją zadań laboratoryjnych zrealizowanych w ramach ćwiczenia nr 8 z wykorzystaniem tego samego oprogramowania. Jego celem jest zapoznanie się z kolejnymi elementami składowymi oprogramowania InTouch oraz nabycie umiejętności posługiwania się nimi.

Zakres tego ćwiczenia ograniczony jest do nabycia umiejętności definiowania zmiennych, tworzenia połączeń animacyjnych oraz pisanie prostych skryptów.

Wymagane przygotowanie:

1. Zrealizowane zadania z ćwiczenia nr 8.
2. Znajomość zasad definiowania zmiennych (patrz: [1] s. 283-454).
3. Znajomość zasad tworzenia połączeń animacyjnych (patrz: [1] s. 455-504).
4. Znajomość zasad tworzenia skryptów (patrz: [1] s. 505-584).

Literatura

1. Wonderware InTouch - Podręcznik użytkownika (dostępny on-line: http://www.astor.com.pl/page/downloads/archive/Centrum_techiczne/Wonderware/dokumentacja_pl/LI-ASK-PU-IT95.pdf)

II. ZADANIA LABORATORYJNE

Zadanie 1. Zapoznanie się ze sposobem definiowania zmiennych

W trakcie realizacji zadania należy:

Korzystając z WindowMaker'a otworzyć aplikację stworzoną podczas wykonywania ćwiczenia 8, a następnie dokonać deklaracji określonej liczby zmiennych, zgodnie z poniższym opisem:

1. Zapoznać się z oknem deklaracji zmiennych (w celu przywołania okna skorzystać z menu **Special/ Tagname Dictionary**).
2. Zdefiniować zmienne wymienione w poniższej tabeli:

Lp	Nazwa zmiennej (tagname)	Typ zmiennej (Type)	Wartość początkowa (Initial value)	Wartość minimalna (Min Value)	Wartość maksymalna (Max Value)	Opis funkcji pełnionej przez zmienną
1.	zawór1	Memory Discrete	Off	-	-	odzwierciedlenie stanu zaworu wlewowego do zbiornika
2.	zawór2	Memory Discrete	Off	-	-	odzwierciedlenie stanu zaworu wylotowego ze zbiornika

3.	przełącznik	Memory Discrete	Off	-	-	ilustracja sposobów zmiany zmiennej
4.	dysk	Memory Integer	0	0	9999	ilość wolnego miejsca na dysku twardym
5.	pamięć	Memory Integer	0	0	999999	ilość wolnej pamięci pod Windows
6.	poziom	Memory Integer	60	0	100	poziom medium w zbiorniku
7.	procent	Memory Integer	0	0	100	stopień otwarcia zaworu wylotowego
8.	kąt	Memory Integer	90	0	360	kąt obrotu
9.	sinus	Memory real	1	-1	1	sinus kąta obrotu

Zadanie 2. Połączenia animacyjne

W trakcie realizacji zadania należy:

Dodać połączenia animacyjne do wybranych obiektów utworzonych w ćwiczeniu 8, wykorzystując zdefiniowane zmienne, zgodnie z poniższym opisem:

a) Utworzyć w oknie „Menu” następujące połączenia animacyjne, wyświetlające określone okna po naciśnięciu odpowiednich przycisków:

1. Dla przycisku o etykiecie „Data i czas” – podać do wyświetlenia okno „Data i czas”.

2. Dla przycisku o etykiecie „Animacja” – podać do wyświetlenia okno „Animacja”.

Zapisać zmiany wprowadzone w oknie „Menu”.

b) Utworzyć w oknie „Data i czas” następujące połączenia animacyjne:

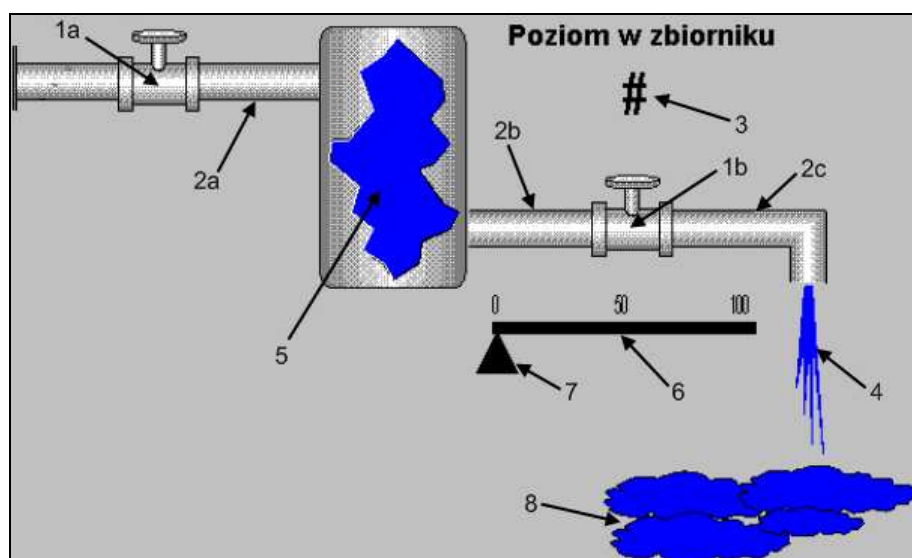
1. Dla obiektu tekstowego „# MB” – połączenie (**Value Display/ Analog**) wiążące ten obiekt z wartością zmiennej określającej ilość wolnego miejsca na dysku twardym.

2. Dla obiektu tekstowego „# KB” – połączenie wiążące ten obiekt z wartością zmiennej określającej ilość wolnej pamięci pod Windows.

3. Dla przycisku „Menu” – wykorzystując zmienną systemową „\$AccessLevel” ustawić poziom widoczności obiektu na wartość powyżej 9000, tak aby dla wartości większej od 9000 obiekt ten był widoczny.

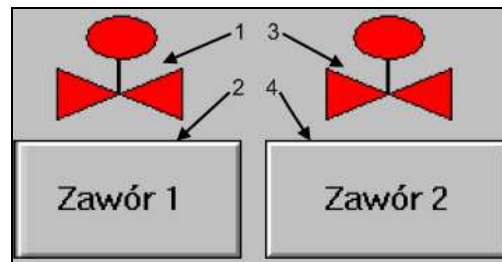
Zapisać zmiany wprowadzone w oknie „Data i czas”.

c) Przyjmując numerację obiektów jak na rysunku 1, utworzyć w oknie „Animacja” następujące połączenia animacyjne:



Rysunek 1 Numeracja obiektów w oknie „Animacja” – część pierwsza.

1. **Obiekt 1a** – dyskretne połączenie dotykowe (**Touch Links/ Touch Pushbuttons/ Discrete Value**) wiążące obiekt ze zmienną odzwierciedlającą stan zaworu wlewowego (tzn. „zawór1”). Ustawić właściwości obiektu tak, aby zawór był otwarty tylko w momencie, gdy zostanie on zaznaczony poprzez kliknięcie na nim i przytrzymanie lewego przycisku myszy. W przeciwnym przypadku zawór musi pozostać zamknięty. Powiązać kolor linii obiektu ze stanem zaworu. Ustawić: kolor niebieski, gdy zawór jest otwarty (tzn. gdy zmienna „zawór1” przyjmuje wartość „True”) oraz czarny gdy zawór jest zamknięty (tzn. jeśli zmienna „zawór1” przyjmuje wartość „False”).
 2. **Obiekt 1b** – wykonać polecenie analogicznie jak w punkcie 1., z tym że powiązania obiektu dokonać ze zmienną określającą stan zaworu wylotowego („zawór2”).
 3. **Obiekt 2a** – dyskretne połączenie dotykowe (**Animation.../ Line color/ Discrete**) wiążące kolor linii obiektu ze stanem zaworu wlotowego. Kolor linii ustawić analogicznie jak w punkcie 1.
 4. **Obiekt 2b** – powiązać połączeniem dyskretnym kolor linii obiektu ze zmienną odzwierciedlającą poziom medium w zbiorniku. Ustawić kolor na niebieski, gdy ciecz będzie uchodzić ze zbiornika (a więc np. gdy wartość zmiennej „poziom” będzie większa od 25 – dokładna wartość tej zmiennej zależna jest od poziomu, na którym umieszczony jest wylew ze zbiornika), w przeciwnym przypadku przypisać kolor czarny.
 5. **Obiekt 2c** – powiązać połączeniem dyskretnym kolor linii ze stanem zaworu wylotowego, stopniem jego otwarcia oraz poziomem medium w zbiorniku. Ustawić kolor linii odpowiednio: niebieski w przypadku przepływu cieczy oraz czarny w przeciwnym przypadku.
 6. **Obiekt 3** – połączenie wyświetlające w obiekcie tekstowym poziom wody w zbiorniku (tzn. wiążące ten obiekt ze zmienną „poziom” – **Touch Links/User Inputs/ Analog**).
 7. **Obiekt 4** – uzależnić wysokość oraz szerokość obiektu od wartości poziomu cieczy w zbiorniku oraz poziomu otwarcia zaworu wylotowego. Ustawić zakotwiczenie w pozycji „Top” dla wysokości obiektu oraz w pozycji „Center” dla szerokości. Ustawić warunek widoczności obiektu korzystając z przykładowego skryptu: **zawór2 AND procent > 0 AND poziom > 25**. Ustawić pole opcji „Stan widoczny” (Visible state) na „On”. Jaka jest interpretacja podanego skryptu?
 8. **Obiekt 5** – połączenie wiążące poziom wypełnienia pionowego obiektu (**Animation.../ Percent Fill/ Vertical**) z ilością cieczy w zbiorniku.
 9. **Obiekt 6** – uzależnić poziom wypełnienia poziomego obiektu od stopnia otwarcia zaworu wylotowego.
 10. **Obiekt 7** – uzależnić stopień otwarcia zaworu wylotowego od przesunięcia poziomego suwaka. Ustawić minimalne otwarcie zaworu przy maksymalnym przesunięciu suwaka w lewo oraz pełne otwarcie przy maksymalnym wysunięciu suwaka w prawo (wartość maksymalnego wysunięcia powinna być równa szerokości obiektu 6). Zmienić ustawienia obiektu tak, aby był on nieaktywny w momencie, gdy zawór wylotowy jest zamknięty (tzn. zmienna „zawór2” przyjmuje wartość „False”).
 11. **Obiekt 8** – uzależnić widoczność obiektu od poziomu cieczy w zbiorniku, stopnia otwarcia zaworu wylotowego oraz stanu zaworu wylotowego.
Zapisać zmiany wprowadzone w oknie „Animacja”.
- d) Przyjmując numerację obiektów jak na poniższym rysunku 2, utworzyć w oknie „Animacja” następujące połączenia animacyjne:

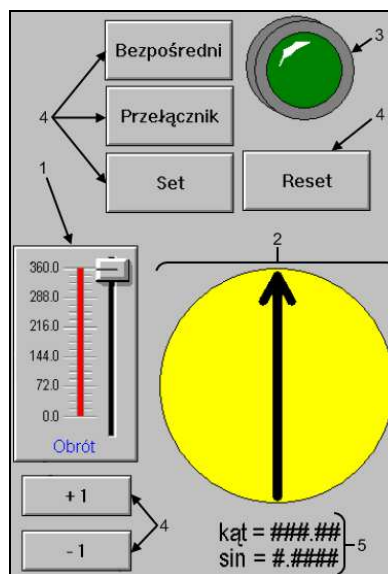


Rysunek 2 Numeracja obiektów w oknie „Animacja” – część druga.

1. **Obiekt 1** – uzależnić kolor obiektu od stanu zaworu wlewowego do zbiornika, korzystając z wartości zmiennej dyskretnej „zawór1”. Przypisać: kolor niebieski, gdy zawór jest otwarty oraz kolor czarny, gdy zawór jest zamknięty.
2. **Obiekt 2** – utworzyć dyskretne połączenie dotykowe pomiędzy przyciskiem o etykiecie „Zawór 1” oraz stanem zaworu wlewowego. Ustawić opcje przycisku tak, aby jego jednorazowe wciśnięcie powodowało zmianę stanu zaworu.
3. **Obiekt 3, Obiekt 4** – dokonać operacji analogicznych jak dla obiektów odpowiednio pierwszego oraz drugiego z tym, że w tym przypadku rozważać stan zaworu wylotowego.

Zapisać zmiany wprowadzone w oknie „Animacja”.

- e) Przyjmując numerację obiektów jak na poniższym rysunku 3, utworzyć w oknie „Animacja” następujące połączenia animacyjne:



Rysunek 3 Numeracja obiektów w oknie „Animacja” – część trzecia.

1. **Obiekt 1** – powiązać obiekt (tzn. ustawić wartość właściwości „Tagname”) ze zmienną „kąt”.
2. **Obiekt 2** – zgrupować poszczególne elementy strzałki w symbol, a następnie przy pomocy opcji **Miscellaneous/ Orientation** uzależnić obrót obiektu od wartości zmiennej kąta obrotu.
3. **Obiekt 3** – Uzależnić kolor obiektu od wartości zmiennej dyskretnej „przełącznik”. Ustawić: kolor ciemnozielony, gdy wartość tej zmiennej wynosi „False”, w przeciwnym przypadku przypisać kolor jasnozielony.
4. **Obiekty nr 4** – utworzyć dyskretne połączenie pomiędzy przyciskami „Bezpośredni”, „Przełącznik”, „Set”, „Reset” oraz zmienną „przełącznik”. Ustawić dla poszczególnych przycisków wartość właściwości „Action” kolejno na „Direct”,

„Toggle”, „Set”, „Reset”. Zastanowić się nad różnicami w funkcjonowaniu przycisków uwzględniając poszczególne ustawienia.

5. **Obiekty nr 5** – powiązać teksty „ką = ###.##” oraz „sin = #.####” ze zmiennymi odpowiednio kąta obrotu oraz sinusa kąta obrotu.

Zapisać zmiany wprowadzone w oknie „Animacja”.

- f) Uruchomić aplikację WindowViewer korzystając z menu „Runtime”, a następnie przetestować wykonane połączenia animacyjne.

Zadanie 3. Tworzenie skryptów.

W trakcie realizacji zadania należy:

Wprowadzić przedstawione poniżej skrypty zgodnie z poniższym opisem, przeanalizować ich działanie, a następnie napisać własny skrypt.

1. Okno „Data i czas”: Utworzyć skrypt typu „Windows Script” wykonywany co 60 sekund, wtedy gdy okno to jest widoczne.

Treść skryptu:

```
dysk=InfoDisk( "C", 2, $Minute)/(1024*1024);
{co minutę wyliczanie ilości wolnego miejsca na dysku C w MB}
pamięć=InfoResources( 3, $Minute)/1024;2
{co minutę wyliczanie ilości wolnej pamięci pod Windows w KB}
```

2. Okno „Menu”, przycisk „Kalkulator”: Utworzyć skrypt, który zostanie wykonany w momencie wciśnięcia przycisku.

Treść skryptu:

```
IF InfoAppActive("Kalkulator")==1 THEN
{sprawdzenie czy Kalkulator jest w pamięci}
ActivateApp "Kalkulator";
{jeśli tak - uruchomienie go z pamięci}
ELSE
StartApp "calc.exe";
{jeśli nie - uruchomienie z dysku}
ENDIF;
```

3. Okno „Menu”, przycisk „Koniec”: Utworzyć skrypt, który zostanie wykonany w momencie wciśnięcia przycisku.

Treść skryptu:

```
WWControl( InfoAppTitle( "view" ), "Close" );
```

4. Okno „Animacja”, przycisk „+ 1”: Utworzyć skrypt, który zostanie wykonany co sekundę w momencie wciśnięcia/wciśnięcia i przytrzymania lewego przycisku myszy.

Treść skryptu:

```
ką = ką + 1;
```

5. Okno „Animacja”, przycisk „- 1”: Utworzyć skrypt, który zostanie wykonany co sekundę w momencie wciśnięcia/wciśnięcia i przytrzymania lewego przycisku myszy.

Treść skryptu:

```
ką = ką - 1;
```

6. Skrypt typu „Data Change Script”. Ustawić opcje – Tagname: ką

Treść skryptu:

```
IF kat < 360 THEN
    sinus = Sin(kat);
ENDIF;
```

7. Skrypt typu „Condition Script”. Ustawić opcje – Condition Type: On True, Condition: **kat == 360**

Treść skryptu:

```
sinus = 0;
```

8. Okno „Animacja”: Utworzyć własny skrypt typu „Windows Script”, wykonywany co określony czas (np. sekundę), wtedy gdy okno to jest widoczne.

Zadaniem skryptu jest symulacja pracy zbiornika, zgodnie z następującymi zasadami:

- gdy zawór 1 jest otwarty oraz gdy poziom cieczy w zbiorniku jest niższy niż 100%, wówczas następuje wzrost poziomu cieczy w zbiorniku o 1%,
- gdy zawór 2 jest otwarty oraz gdy poziom cieczy w zbiorniku jest wyższy od zadanego (np. 24%), wówczas następuje obniżenie poziomu cieczy w zależności od stopnia otwarcia zaworu wylotowego, opisanego zmienną *procent* (np. o $0,2 * procent$).

Wyniki

Student demonstruje prowadzącemu umiejętność definiowania zmiennych, tworzenia połączeń animacyjnych, a także pisania skryptów,

Na koniec zajęć, student dostarcza prowadzącemu zajęcia w formie elektronicznej aplikację, w której wykonane zostały zadania nr 1, 2 oraz 3.