Politechnika Poznańska
Wydział Budowy Maszyn
i Zarządzania

Automatyzacja i Nadzorowanie Maszyn
Zajęcia laboratoryjne

Ćwiczenie 10
Wizualizacja

24.04.2018

Poznań 2017

OGÓLNE ZASADY BEZPIECZEŃSTWA
PODCZAS WYKONYWANIA ĆWICZEŃ LABORATORYJNYCH

! Przed przystąpieniem do ćwiczenia należy zapoznać się z instrukcją dydaktyczną.

! Dokonać oględzin urządzeń, przyrządów i przewodów używanych podczas ćwiczenia. W przypadku zauważenia nieprawidłowości lub uszkodzeń bezzwłocznie powiadomić prowadzącego.

! Zabrania się samodzielnego załączania stanowiska bez sprawdzenia połączeń i wydaniu zgody przez prowadzącego.

! Zmian parametrów lub konfiguracji stanowiska przy użyciu dostępnych przełączników i potencjometrów można dokonywać po uprzednim przeanalizowaniu skutków takich działań.

! Zmian w konfiguracji obwodów elektrycznych polegających na zmianie połączeń przewodów lub wymianie przyrządów, należy dokonywać po uprzednim wyłączeniu zasilania stanowiska.

! Zabrania się wykonywania przełączeń (przewodów, urządzeń) w układzie znajdującym się pod napięciem.

! Przy obsłudze stanowisk, które zawierają elementy zasilane napięciem elektrycznym wyższym niż napięcie bezpieczne, należy zachować szczególną ostrożność w celu uniknięcia porażenia prądem elektrycznym.

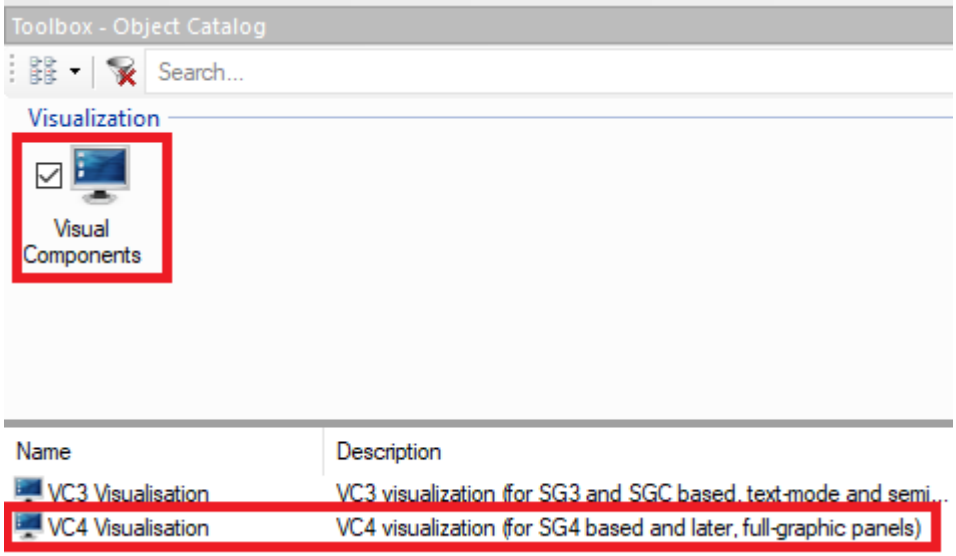
! Stosowanie ustawień i procedur innych niż opisane w instrukcji lub zalecone przez prowadzącego może spowodować nieprzewidziane działanie, a nawet uszkodzenie stanowiska.

! Przekroczenie dopuszczalnych parametrów (napięć, prądów) może doprowadzić do uszkodzenia elementów stanowiska, pożaru lub porażenia prądem.

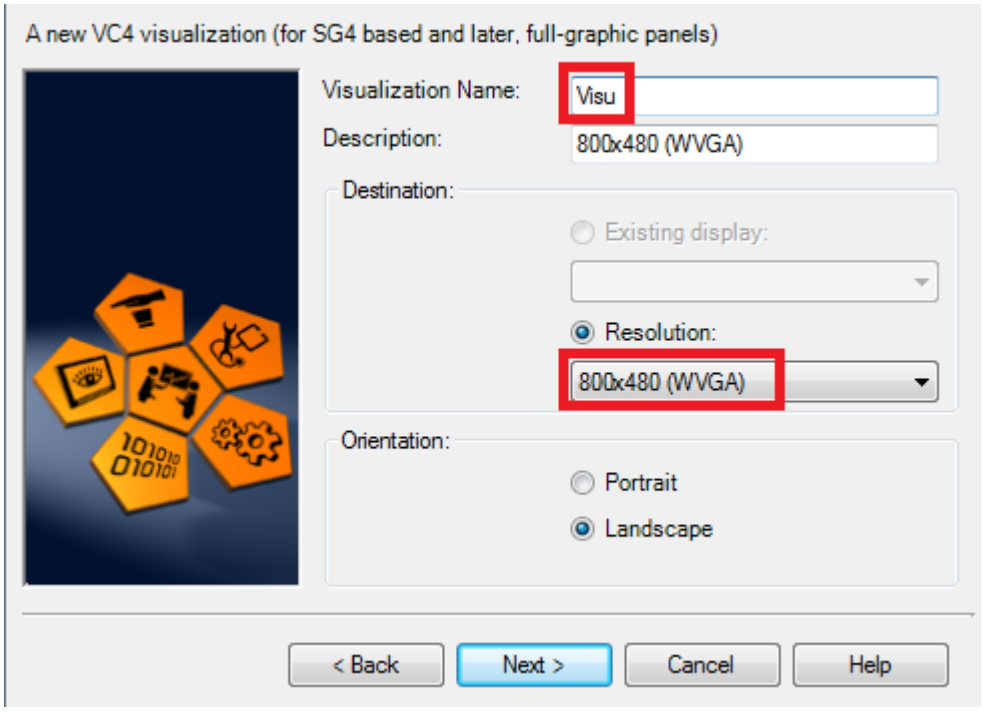
! W przypadku nieprawidłowego działania urządzeń lub wystąpienia objawów uszkodzeń (np. iskrzenie, zapach spalenizny) należy natychmiast wyłączyć stanowisko i powiadomić prowadzącego.

1. Dodawanie wizualizacji

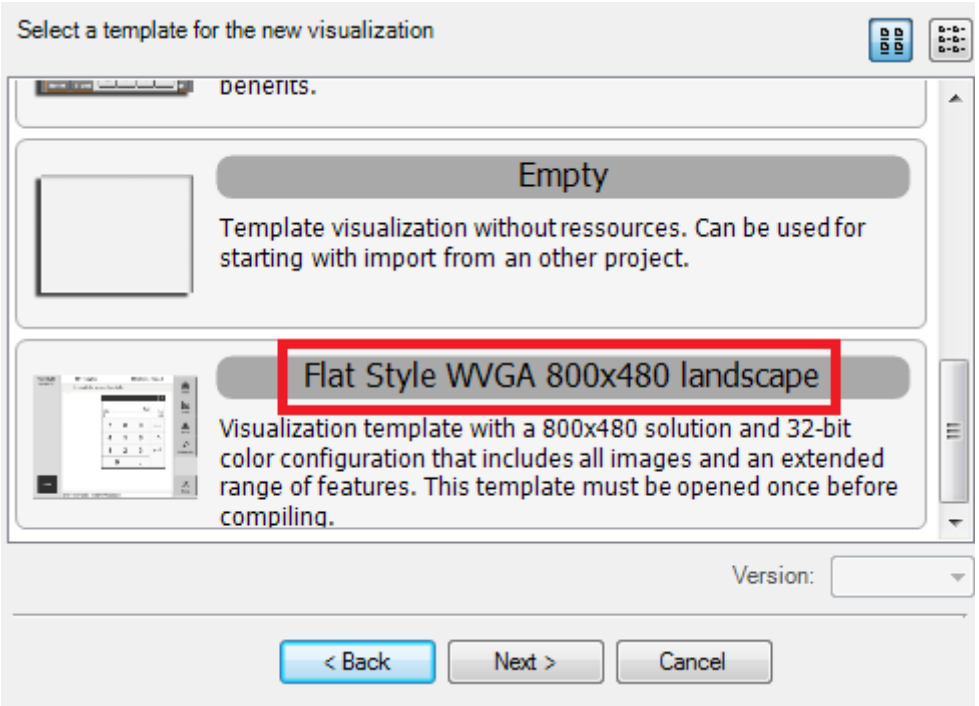
W pierwszym kroku tworzymy projekt (opis tworzenia projektu znajduje się w **poprzedniej** instrukcji!) a następnie kursorem wybieramy zakładkę Logical View . Następnie wybieramy lewym klawiszem myszy na nazwie projektu. W Toolboxie zaznaczyć należy **Visual Components** i kliknać 2 razy w **VC4 Visualisation**.



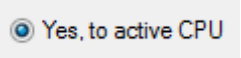
W kolejnym oknie podajemy nazwę wizualizacji i wybieramy rozdzielczość 800x480 (WVGA). Pozostałe ustawienia bez zmian i klikamy przycisk Next >.



W następnym oknie wybieramy szablon Flat Style WVGA 800x480 landscape.

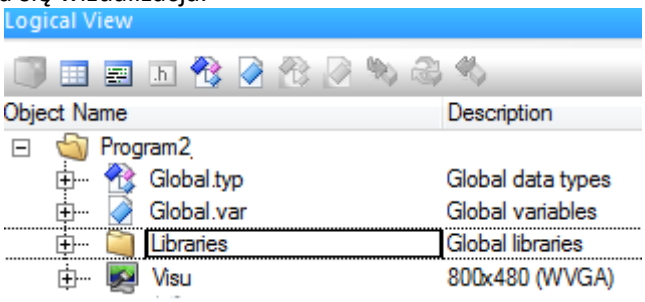


I klikamy przycisk Next >. W ostatnim oknie zaznaczamy opcję



Kończymy przyciskiem Finish.

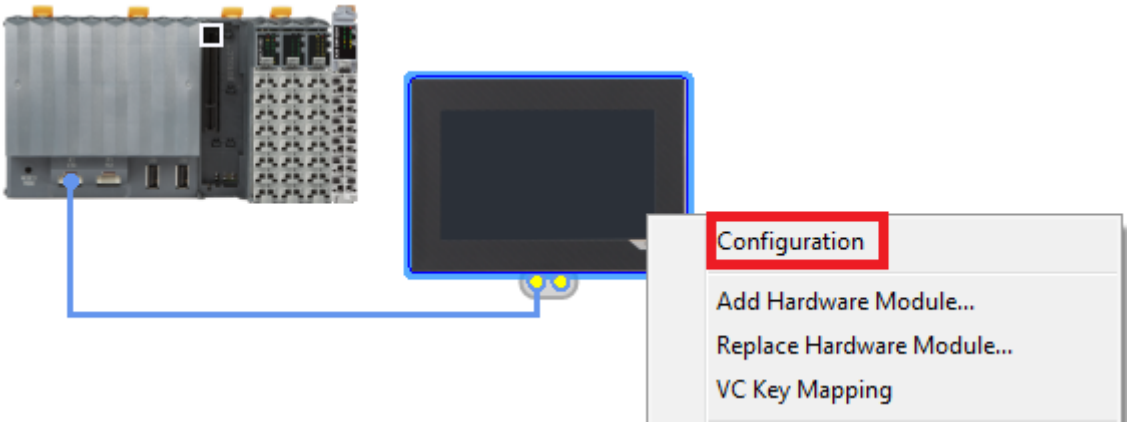
W oknie projektu dodała się wizualizacja.



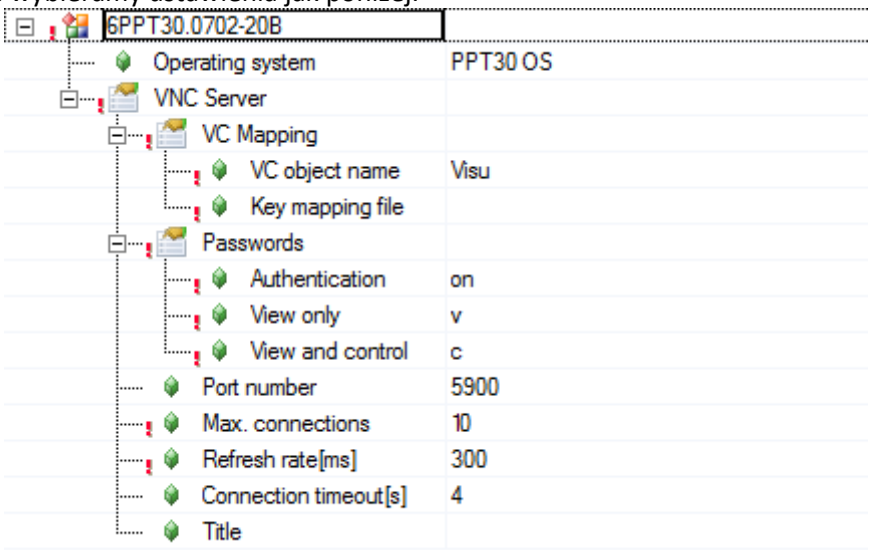
1.1. Podpinanie wizualizacji

1.1.1.Stanowiska z panelem

Przechodzimy do System Designera klikając ikonę , a następnie prawym przyciskiem myszy na dodany wcześniej panel i wybieramy Configuration.

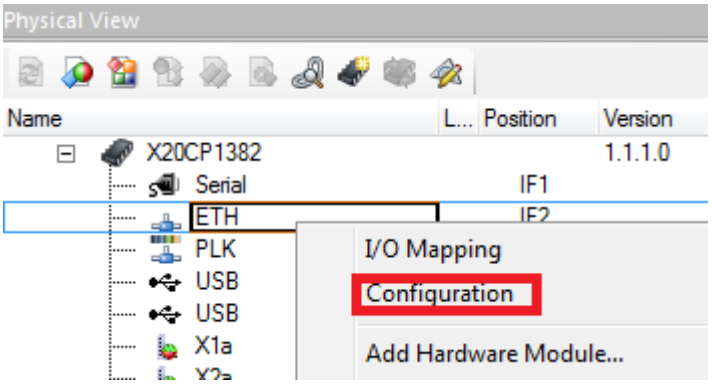


W konfiguracji wybieramy ustawienia jak poniżej.

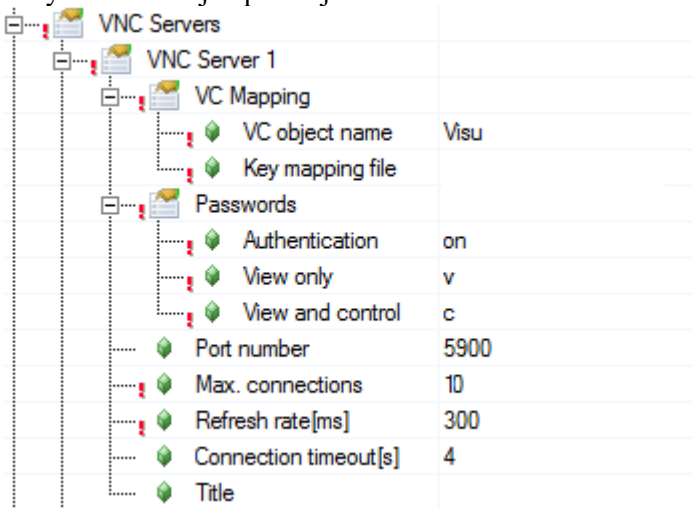


1.1.2. Stanowiska z silnikiem

Klikamy na zakładkę Physical View , a następnie prawym przyciskiem myszy na ETH i wybieramy Configuration.



W konfiguracji wybieramy ustawienia jak poniżej.

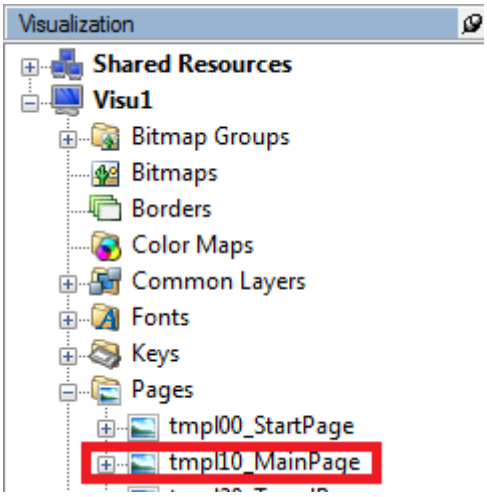


2. Obsługa przycisku na wizualizacji

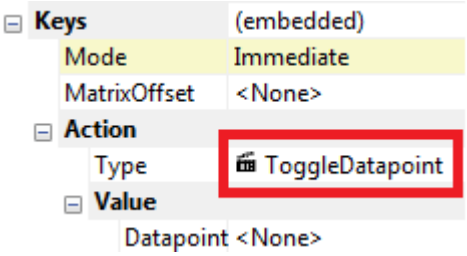
Tworzymy program w języku ST, w którym przepisujemy kod poniżej w części **INIT** i tworzymy zmienną **wejscie** typu **BOOL**.

```
PROGRAM _INIT

    wejscie:=0;

END_PROGRAM
```

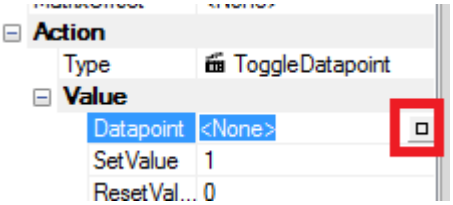
W następnym kroku klikamy dwa racy lewym przyciskiem myszy na wcześniej dodaną wizualizację. Z listy rozwijanej wybieramy Pages -> tmpl10_MainPage.



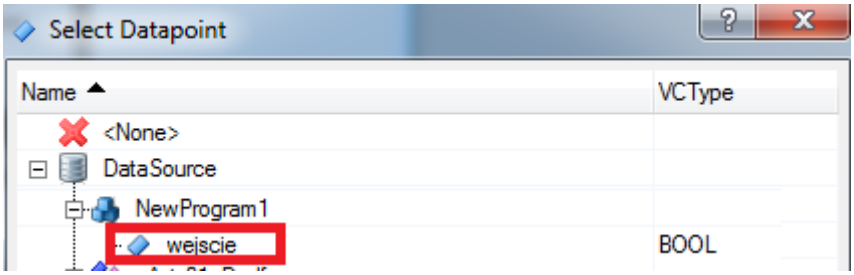
Z zakładki Tools wybieramy przycisk  i przenosimy go na otwartą podstronę i dostosowujemy jego wielkość . Zaznaczamy go i po prawej stronie w zakładce Properties w sekcji Action zmieniamy na ToggleDatapoint.



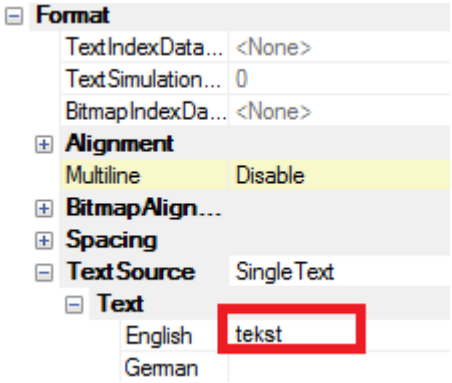
W części Value wybieramy Datapoint.



Teraz z listy wybieramy zmienną **wejscie**.



Aby zmienić napis na przycisku z sekcji Properties wybieramy

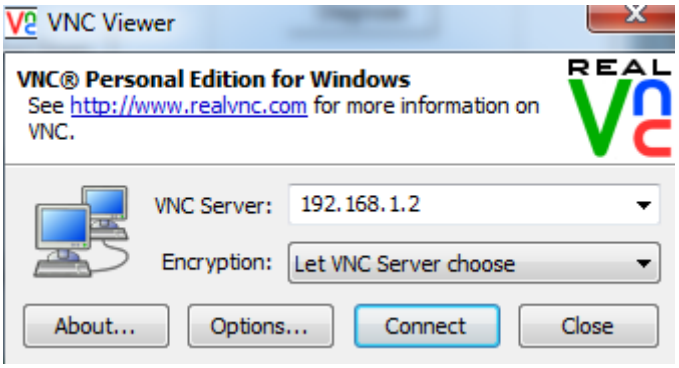


Ostatnim krokiem jest skompilowanie programu i wgranie go na sterownik.

2.1. Stanowiska z silnikiem

Osoby na stanowiskach z silnikiem uruchamiają program

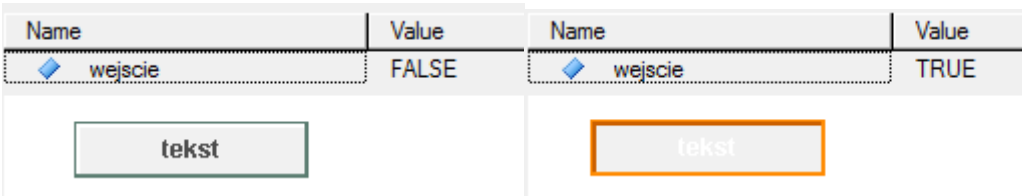
Następnie wpisują adres IP sterownika



Następnie kliknąć przycisk Connect. W polu password wpisać **c** i kliknąć przycisk OK.

2.2. Sprawdzenie działania programu

Sprawdzić działanie programu klikając na przycisk i sprawdzając czy zmienna w Watchu się zmienia

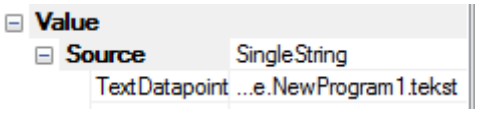


3. Zmiana tekstu na wizualizacji

W pierwszym kroku do poprzedniego programu dopisujemy w części **CYCLIC** kod poniżej. Zmienna **tekst** zdefiniowana jako **STRING[20]**.

```
PROGRAM _CYCLIC

    IF wejscie=0 THEN
        tekst:='TEKST 1';
    ELSE
        tekst:='TEKST 2';
    END_IF;

END_PROGRAM
```

Na wizualizacji wybieramy przycisk  i przeciągamy na wizualizację. W zakładce Properties wybieramy Source jako SingleString. Komórka niżej wybieramy zmienną **tekst**.



Po kompilacji programu sprawdzamy na wizualizacji czy wraz z wciśnięciem przycisku zmienia się tekst.

4. Obsługa klawiatury numerycznej

W pierwszej kolejności edytujemy program jak poniżej. Zmienne **wejście_int** oraz **wyjście_int** definiujemy jako zmienne **INT**.

```
PROGRAM _CYCLIC

    IF wejscie_int>=1000 THEN
        tekst:='TEKST 1';
    ELSE
        tekst:='TEKST 2';
    END_IF;

    wyjscie_int:=wejscie_int*2;

END_PROGRAM
```

Następnie wybieramy przycisk 123 i przeciągamy dwukrotnie na stronę wizualizacji. W pierwszym ustawiamy zmienną **wejście_int**, a w drugim **wyjście_int**.

Name	Numeric_1
StyleClass	Default
⊞ **Position**	
⊞ **Appearance**	
⊞ **Format**	
⊟ **Value**	Standard
Datapoint	<None>
MinValue	<None>

W oknie gdzie podpięliśmy zmienną **wejście_int** ustawiamy parametry **MinValue** na 0 i **MaxValue** na 2000. W kolejnym kroku zmieniamy parametr **Input** na **TRUE** i zmieniamy patametr **TouchPad** na **NumPad**.

⊟ **Input**	True
Start	Any Key
Confirm	Enter
Next	Enter
Cancel	Lost Focus
Comma	Explicit
Minus	Normal
UpDown	Disabled
TouchPad	NumPad
⊞ **Runtime**	

Po skompilowaniu programu i wysłaniu go na sterownik klikamy na panelu na pierwsze okno. Powinno pojawić się okno z klawiaturą numeryczną. Po wpisaniu wartości w drugim oknie pojawi się wartość pomnożona razy 2. Jeśli wpiszemy wartość powyżej lub równą 1000 powinien zmienić się wyświetlany tekst.

5. Zmiana koloru

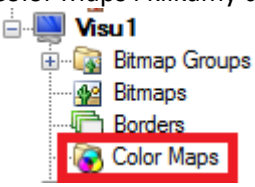
W pierwszej kolejności edytujemy program jak poniżej. Zmienna **kolor** jest zmienną **INT.**

```
PROGRAM _CYCLIC

    IF wejscie_int>=1000 THEN
        tekst:='TEKST 1';
        kolor:=0;
    ELSE
        tekst:='TEKST 2';
        kolor:=1;
    END_IF;

    wyjscie_int:=wejscie_int*2;

END_PROGRAM
```

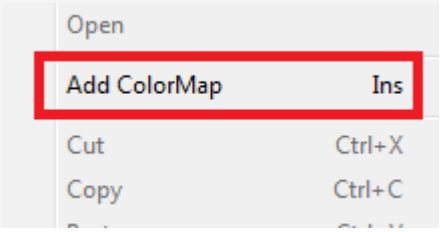
Następnie wybieramy przycisk i przeciągamy na stronę wizualizacji. Zmieniamy kształt na elipsę.

⊟ **Format**	
LineWidth	1
Shape	Ellipse

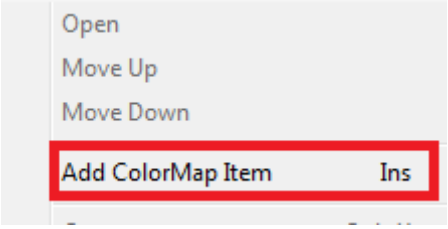
W następnym kroku wybieramy z listy Color Maps i klikamy dwukrotnie lewym przyciskiem myszy.



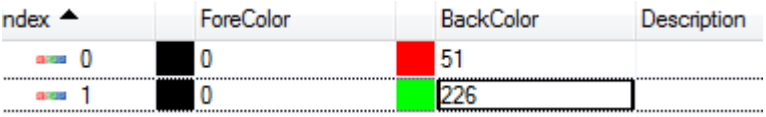
W oknie klikamy prawym klawiszem myszy i wybieramy Add ColorMap.



W kolejnym oknie klikamy prawym klawiszem myszy i wybieramy dwa razy Add ColorMap Item.

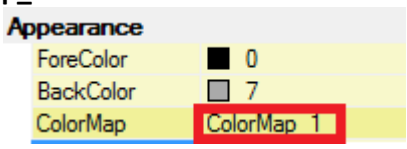


Kolory ustawiamy tak jak na zdjęciu poniżej.



ndex	ForeColor	BackColor	Description
0	0	51	
1	0	226	

Teraz wracamy do okna z podstroną wizualizacji. Klikamy na nasz okrąg i w zakładce Properties zmieniamy ColorMap na **ColorMap_1**.



Poniżej ustawiamy Color Datapoint na zmienną **kolor**. Teraz pozostaje skompilować program i wgrać go do sterownika. Następnie sprawdzamy czy zmienia się kolor wypełnienia okręgu jeśli wpiszemy w oknie z poprzedniego przykładu wartość większą od 1000.

6. Zadanie dla studentów

Student ma za zadanie bazując na poprzednich przykładach stworzyć wizualizację sygnalizacji drogowej. Program obsługujący zmianę świateł przedstawiono poniżej. Dodatkowo należy dodać przycisk włączający światła.

```
IF stan=0 AND start=1 THEN
stan:=1;
ELSIF start=0 THEN
stan:=0;
END_IF;


CASE stan OF
    0:
        zielone:=0;
        zolte:=0;
        czerwone:=0;

        TON_Zielone.IN:=0;
        TON_Zolte.IN:=0;
        TON_Czerwone.IN:=0;
    1://zielone swiatlo
        zielone:=1;
        zolte:=0;
        czerwone:=0;
        TON_Zielone.PT:=T#10s;
        TON_Zielone.IN:=1;
        IF TON_Zielone.Q THEN
            stan:=2;
            TON_Zielone.IN:=0;
        END_IF;
    2://zolte swiatlo
        zielone:=0;
        zolte:=1;
        czerwone:=0;
        TON_Zolte.PT:=T#1s;
        TON_Zolte.IN:=1;
        IF TON_Zolte.Q THEN
            stan:=3;
            TON_Zolte.IN:=0;
        END_IF;
    3://czerwone swiatlo
        zielone:=0;
        zolte:=0;
        czerwone:=1;
        TON_Czerwone.PT:=T#10s;
        TON_Czerwone.IN:=1;
        IF TON_Czerwone.Q THEN
            stan:=4;
            TON_Czerwone.IN:=0;
        END_IF;
```

```
    4://zolte swiatlo
        zielone:=0;
        zolte:=1;
        czerwone:=1;
        TON_Zolte.PT:=T#1s;
        TON_Zolte.IN:=1;
        IF TON_Zolte.Q THEN
            stan:=1;
            TON_Zolte.IN:=0;
        END_IF;
END_CASE;

TON_Zielone();
TON_Zolte();
TON_Czerwone();
```

Tabela ze zmiennymi jest podana poniżej.

start	BOOL
stan	INT
zielone	BOOL
zolte	BOOL
czerwone	BOOL
TON_Zielone	TON
TON_Zolte	TON
TON_Czerwone	TON