

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie			
Laboratorium Aparatury Automatykacji			
Numer i temat ćwiczenia: Ćwiczenie 8. Pakiet VisiDaq do realizacji bezpośredniego sterowania cyfrowego			
Grupa ćwiczeniowa: Wtorek 17:00-19:15, Zespół: 3			
Lp.	Imię i nazwisko	Ocena	Podpis
1.	Katarzyna Wątorska		
2.	Sonia Wittek		
3.	Karolina Świerczek		
Data wykonania ćwiczenia: 14.05.2019			

1. Opis pakietu VisiDaq:

VisiDaq to środowisko umożliwiające budowanie aplikacji wykorzystywanych w systemie akwizycji i kontroli danych. Dzięki intuicyjnemu interfejsowi, program jest dość łatwy w obsłudze; graficzne bloki w Task Designerze pozwalają na budowanie strategii bez konieczności konwencjonalnego programowania. Display Designer umożliwia zaprojektowanie panelu użytkownika. Automatycznie generowane raporty przez Report Designer usprawniają wykrywanie awarii. VisiDaq może przeprowadzać złożone obliczenia i analizy. Do jego zalet należy również możliwość integracji z innymi aplikacjami w celu wymiany danych w czasie rzeczywistym.

Środowisko działa tylko w systemie Windows, co jest jego niewątpliwą wadą – zarówno ze względu na zawodność systemu, jak i ograniczoność zastosowania. Fakt, że do programowania jest czasem wykorzystany Visual Basic, który jest językiem interpretowanym, sprawia, że działanie programu może być wolniejsze.

2. Opis stanowiska:

Celem ćwiczenia było zapoznanie się z oprogramowaniem VisiDaq, które pozwalało na realizację sterowania cyfrowego przy użyciu komputera PC oraz zapoznanie się z przykładowym interfejsem procesowym.

Stanowisko składało się z komputera PC z zainstalowanym pakietem VisiDaq, stacji akwizycji danych ADAM5000 z modułami wejścia/wyjścia: ADAM 5060, ADAM 5018 i ADAM 5013; oraz podpiętymi do nich czujnikami temperatury i natężenia światła oraz lampy.

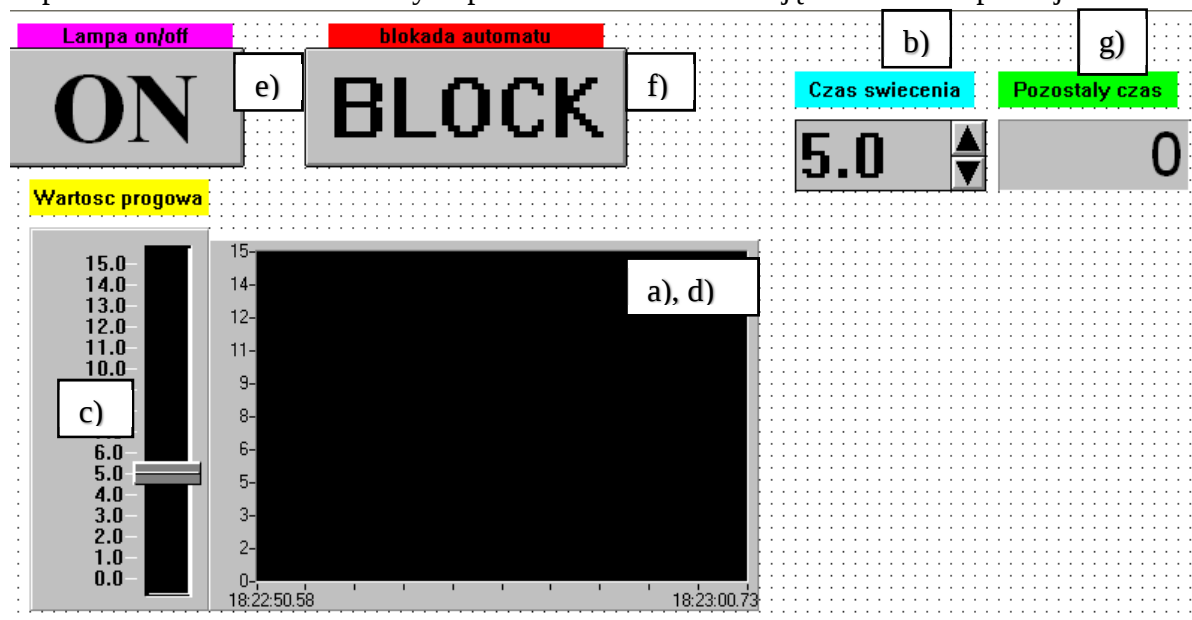
3. Przebieg ćwiczenia:

Podczas laboratorium miałyśmy za zadanie zbudować zaawansowany wyłącznik zmierzchowy, który miał realizować następujące punkty:

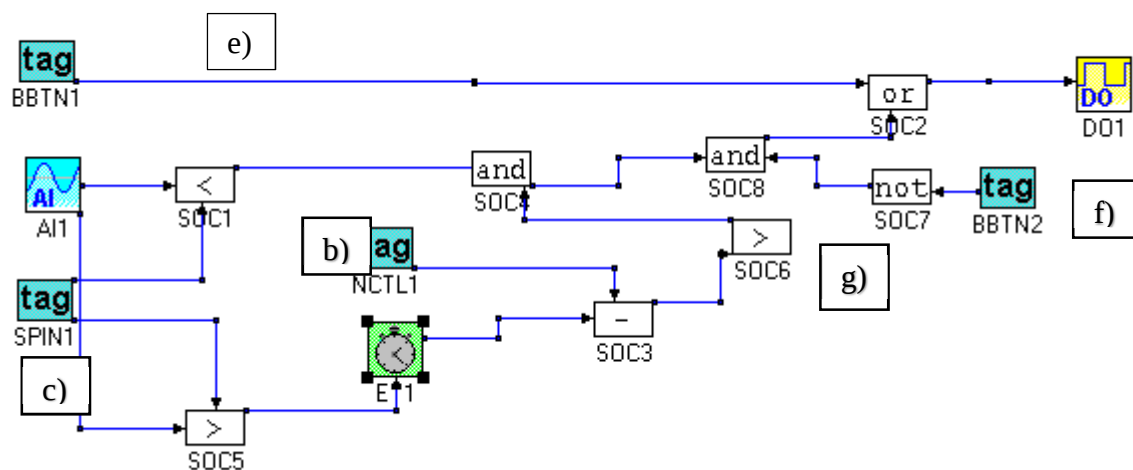
- Pomiar i wyświetlanie na ekranie aktualnego poziomu oświetlenia, wizualizacja czasowego przebiegu aktualnego poziomu oświetlenia
- Zadawanie czasu, na który ma się włączyć lampa
- Włączenie lampy na pewien zadany czas, gdy poziom oświetlenia spadnie poniżej zadanej wartości progowej
- Wizualizacja zadanego progu zadziałania
- Możliwość niezależnego od czujnika włączenia lampy

- f) Możliwość zablokowania automatu włączającego lampę
- g) Wizualizacja czasu pozostałego do wyłączenia lampy przez automat

Aby zrealizować to zadanie otworzyliśmy po jednym oknie Task Designer i Display Designer, w których zaprojektowałyśmy aplikację widoczną poniżej. Na rysunkach zaznaczono odpowiednimi literami elementy odpowiedzialne za realizację elementów aplikacji.



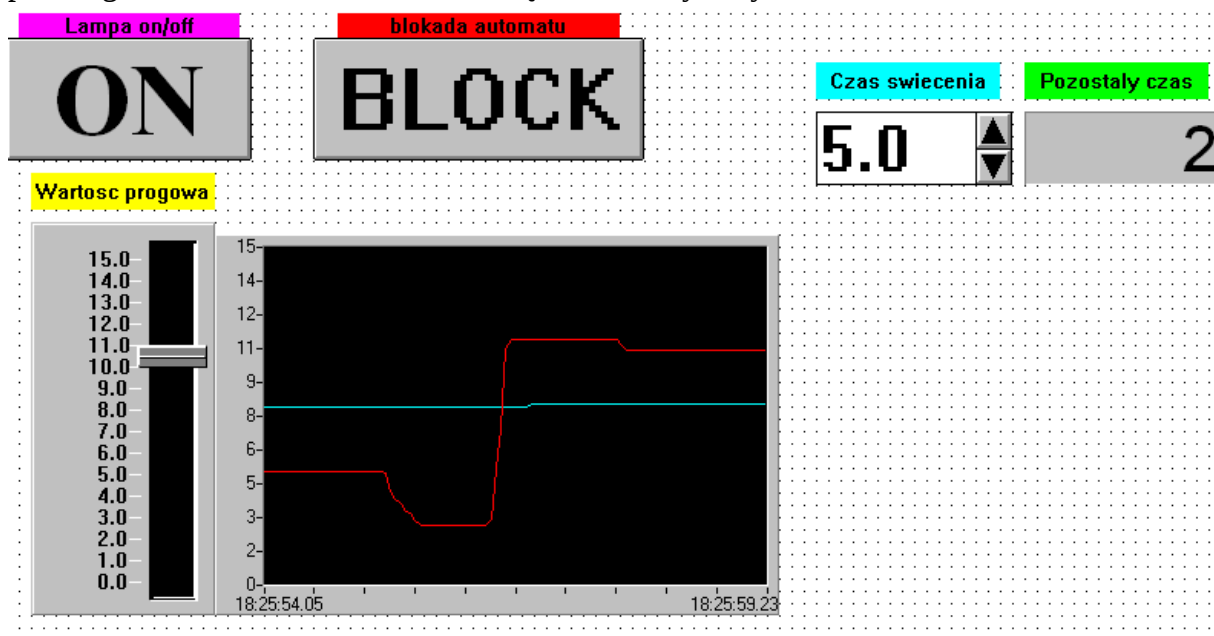
Rysunek 1: Wygląd zaprojektowanej aplikacji w Display Designer.



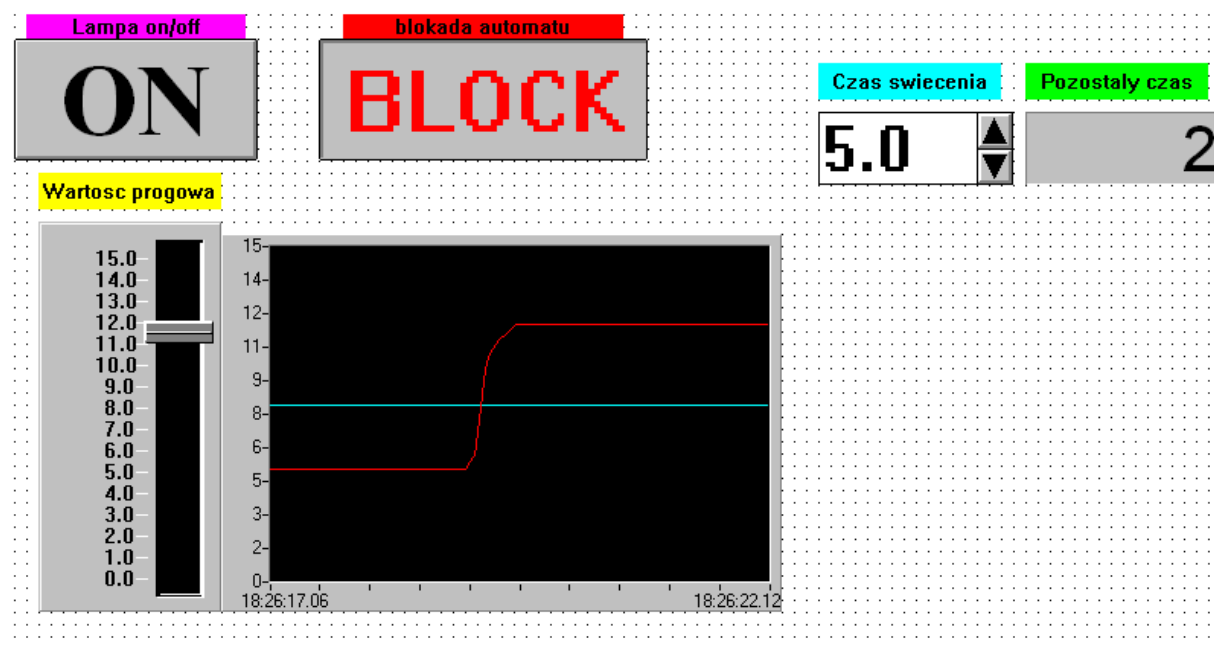
Rysunek 2: Wygląd zaprojektowanej aplikacji w Task Designer.

Gotowa aplikacja działała w oczekiwany sposób; przy zarejestrowaniu włączenia przycisku Lampa on/off (BBTN1 w Task Designer) lampa włączała się (wyjście dyskretne DO1). Po zarejestrowaniu poziomu oświetlenia (Wejście analogowe AI1) i zadanego progu oświetlenia (SPIN1) porównywała je, a jeśli aktualny poziom był niższy od progu włączała lampę oraz odliczanie zegara do końca. Wyłączenie lampy w takiej sytuacji następowało jeśli różnica czasu zegara (ET1) i zadanego czasu (NCTL1) była niższa od 0, sama różnica wyświetlana na panelu operatorskim była pobierana z SOC6. Blokada automatu (BBTN2), gdy włączona, blokowała działanie automatycznego włączania lampy.

Poniżej załączyliśmy zrzuty ekranu obrazujące zachowanie się włączonej aplikacji. Przebieg niebieski na panelu odpowiada zadanemu progowi oświetlenia, a czerwony aktualnemu przebiegowi oświetlenia. Zachowanie się w normalnym trybie:



Rysunek 3: Wygląd panelu aplikacji w normalnym trybie.



Rysunek 4: Wygląd panelu aplikacji po włączeniu blokady automatu.

4. Wnioski:

Podczas tego ćwiczenia dowiedzieliśmy się jak tworzyć proste aplikacje przy użyciu pakietu VisiDaq, jak modyfikować zawartość okien Task Designer, Display Designer oraz jak łączyć zawarte w nich elementy ze sobą, z wejściami lub wyjściami podłączonymi do ADAM-ów. Program, który zaprojektowaliśmy działał poprawnie. Dzięki temu ćwiczeniu nauczyliśmy się pracować ze stacją akwizycji danych ADAM 5000 i modułami ADAM 5060, ADAM 5018, ADAM 5013; była to okazja do poszerzenia wiedzy na ich temat. Dowiedzieliśmy się także, że ze stacją ADAM można porozumieć się poprzez wysyłanie do niej odpowiedniego ciągu znaków ASCII, a aplikacja zaprojektowana na komputerze działa także po odłączeniu od stacji komputera.