



Kierunek studiów Elektrotechnika
Studia II stopnia



Przedmiot:	Instrumentalizacja wirtualna w LabVIEW
Rok:	I
Semestr:	II
Forma studiów:	Studia stacjonarne
Rodzaj zajęć i liczba godzin w semestrze:	
Wykład	30
Ćwiczenia	-
Laboratorium	30
Projekt	-
Liczba punktów ECTS:	5

Cele przedmiotu

C1	Prezentacja koncepcji budowy wirtualnych urządzeń w za pomocą graficznego środowiska programistycznego LabVIEW. Przedstawienie cech charakterystycznych tworzenia instrumentów wirtualnych, metod deklarowania i oznaczania typów danych oraz struktur pozwalających na ich przetwarzanie. Zapoznanie ze sposobem programowania opartym na stosowaniu obiektów graficznych oraz przedstawienie środowiska tworzenia instrumentów wirtualnych.
C2	Prezentowanie zasad tworzenia prostych i złożonych instrumentów wirtualnych (programów). Teoretyczne i praktyczne ćwiczenia śledzenia wykonywania kodu wraz z usuwaniem błędów programu w celu efektywnego wykorzystania środowiska przy rozwiązywaniu problemów z kodem.
C3	Zapoznanie studentów z podstawowymi algorytmami kodu aplikacji oraz metodami lokalnej i zdalnej pracy z instrumentami tworzonymi w środowisku graficznym, z działaniem aplikacji wspierających obsługę sprzętu kontrolno-pomiarowego.
C4	Wprowadzenie w zagadnienia dostosowywania interfejsu programistycznego do potrzeb programisty, indywidualizacji interfejsu aplikacji, dopasowywania aplikacji do możliwości sprzętowych i programowych oraz tworzenia plików wykonywalnych i instalatorów instrumentów wirtualnych.

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1	Swobodne poruszanie się w systemie operacyjnym Windows i posługiwanie się językiem informatyki na poziomie wprowadzanym przez przedmioty takie jak Technologia informacyjna, Podstawy informatyki, Języki programowania oraz znajomość podstawowych pojęć z zakresu metrologii.
----------	---

Efekty kształcenia

	W zakresie wiedzy:
EK 1	Student identyfikuje typy zmiennych, sposób wyróżniania ich w kodzie programu oraz wyjaśnia potencjalne problemy wynikające ze stosowania różnych typów zmiennych.
EK 2	Student rozróżnia techniki i metody odnoszące się do poprawnego wykorzystania struktur zarządzania kodem i wyjaśnia sposób minimalizowania wymagań tworzonych aplikacji w stosunku do systemu operacyjnego, platformy sprzętowej oraz wymagań użytkownika w stosunku do instrumentu wirtualnego.
	W zakresie umiejętności:
EK 3	Student konstruuje proste jednozadaniowe instrumenty wirtualne, bardziej skomplikowane systemy złożone, tworzy pliki wykonywalne i instalacyjne dla stworzonego samodzielnie systemu (np. pomiarowego).
EK 4	Student analizuje wykonywanie kodu programu, identyfikuje występujące błędy i ocenia poprawność stosowania wybranej architektury programu.
EK 5	Student konstruuje instrumenty wirtualne (aplikacje) umożliwiające zdalne i grupowe użytkowanie zasobów.
	W zakresie kompetencji społecznych:
EK 6	Student pracując w zespole określa wymagania wzajemnie współpracujących instrumentów.

EK 7	Student jest odpowiedzialny za minimalizowanie wymagań tworzonych instrumentów wirtualnych w stosunku do systemu operacyjnego i platformy sprzętowej tak, aby system pracował jak najwydajniej.
-------------	---

Treści programowe przedmiot		
Forma zajęć – wykłady		
	Treści programowe	Liczba godzin
W1	Pojęcie instrumentu wirtualnego. Środowisko LabVIEW – charakterystyka, instalacja, panel czołowy, schemat blokowy, palety narzędzi.	2
W2	Typy danych - rozpoznawanie na podstawie symboli i kolorystyki obiektów, zmiana typu danych. Analiza/usuwanie błędów. Metody śledzenia kodu.	2
W3	Modularyzacja - tworzenie, wstawianie, wywoływanie podprogramów.	2
W4	Pętle while i for - zastosowanie, sposób obsługi, tunele danych pętli.	2
W5	Zmienne binarne. Funkcjonowanie przełączników logicznych (mechanical action).	2
W6	Rejestr przesuwany - obsługa obiektu w kodzie programu, zalecane ustawienia. Prezentacja wyników - wskaźniki graficzne (obiekty wykresów waveform chart, waveform graph, XY graph, intensity plot).	2
W7	Macierze /funkcje macierzowe. Klastry /funkcje klastrowe. Rozgałęzianie kodu - struktura wyboru. Wymuszania kolejności wykonywania kodu - struktura sekwencyjna.	2
W8	Dane łańcuchowe - wprowadzanie i wyświetlanie danych tekstowych. Operacje plikowe we/wy z wykorzystaniem zaawansowanych i prostych węzłów środowiska.	2
W9	Indywidualizacja aplikacji i środowiska programistycznego: zasady projektowania panelu czołowego, wprowadzanie klawiszy skrótów.	2
W10	Przyspieszanie pracy programisty dzięki indywidualizacji środowiska programistycznego - modyfikacja właściwości programów, palet, tworzenie własnych obiektów.	2
W11	Planowanie struktury wewnętrznej instrumentu wirtualnego – dobór struktury kodu, projektowanie i wdrażanie mechanizmów obsługi błędów, unikanie nadmiernego wykorzystania procesora i pamięci.	2
W12	Projektowanie panelu czołowego instrumentu – zagadnienia podstawowe, klastry logiczne, programowa obsługa obiektów za pomocą węzłów właściwości.	2
W13	Zdalne sterowanie panelem instrumentu wirtualnego. Udostępnianie danych za pomocą wbudowanego serwera WWW. Konfiguracja klienta zdalnego dostępu do aplikacji.	2
W14	Techniki zarządzania danymi w zakresie jednego instrumentu, wymiany danych w zakresie pojedynczej jednostki, sieciowa wymiana danych – zmienne lokalne i globalne, protokół datasocket.	2
W15	Profilowanie instrumentów wirtualnych (aplikacji). Tworzenie plików wykonywalnych. Generowanie pakietów instalatora.	2
	Suma godzin:	30
Forma zajęć – laboratoria		
	Treści programowe	Liczba godzin
L1	Zajęcia wstępne. Przedstawienie sposobu pracy i zasad obowiązujących w laboratorium. Utworzenie i przetestowanie kont użytkowników. Indywidualne kształtowanie środowiska pracy przez Studenta. Sprawdzenie dostępności zasobów sieciowych.	2
L2	Zapoznanie ze środowiskiem LabVIEW przez stworzenie przyrządu wirtualnego do generowania sygnału i jego prezentacji na panelu czołowym. Korzystanie z szablonów.	2
L3	Edycja elementów panelu czołowego. Korzystanie z węzłów typu Express VI. Ćwiczenie technik usuwania błędów z programu.	2
L4	Pętla While (sposób funkcjonowania, sposób przekazywania danych przez tunele pętli). Rejestr przesuwany. Prezentacja danych za pomocą obiektu Waveform Chart. Stosowanie pętli For.	2
L5	Tworzenie tablic oraz zapoznanie z funkcjami działania na tablicach. Korzystanie z wykresów XY (XY graph). Zapoznanie z korzystaniem z wykresów natężenia (intensity plot). Klastry - tworzenie obiektów klastrow na panelu czołowym oraz korzystanie z funkcji do łączenia i rozłączania danych o charakterze klastrowym.	2

L6	Zapoznanie z wykorzystaniem struktur wyboru. Struktura sekwencyjna - przykładowe zastosowanie. Budowa przyrządu wirtualnego wykorzystującego węzły formuły do wykonywania złożonych działań matematycznych i wyświetlania ich na wykresie.	2
L7	Odrabianie zajęć / Wyrównywanie zaległości.	2
L8	Zmienne łańcuchowe - poznanie funkcji: formatowania do postaci łańcuchowej, łączenia łańcuchów, określania długości łańcuchów, itp. Zapoznanie z mechanizmem obsługi plików z danymi (zapis i odczyt z pliku, zapisywanie tablicy dwuwymiarowej (2D) do pliku tekstowego w postaci arkusza danych.	2
L9	Deklaracja sposobu funkcjonowania podprogramów. Deklarowanie klawiszy skrótu dla funkcji panelu czołowego i konfigurowanie sposobu wyświetlania okien podprogramów inicjowanych za pomocą klawiszy skrótu. Obsługa klastrów za pomocą klawiszy skrótu. Zapoznanie z metodą edycji gotowych programów o konfiguracji utrudniającej modyfikację schematu blokowego.	2
L10	Program (instrument wirtualny) generujący, analizujący i wyświetlający serie danych, wykorzystujący standardowy mechanizm obsługi błędów. Utworzenie programu kontrolującego dane o użytkowniku bazującego na prostym modelu architektury. Zapoznanie z obsługą szablonów dostarczanych ze środowiskiem LabVIEW oraz obsługą szablonów tworzonych samodzielnie.	2
L11	Konfiguracja (optymalizacja) panelu czołowego. Stosowanie kontrolki zakładkowej (tab control). Menu bazujące na klastrze logicznym. Węzły właściwości.	2
L12	Wykorzystanie zmiennych lokalnych do inicjacji, modyfikowania wskaźników i kontrolek panelu czołowego programu. Używanie zmiennych globalnych do wymiany danych pomiędzy programami. Wymiana danych za pomocą mechanizmu DataSocket.	2
L13	Zapis i odczyt danych z plików binarnych. Przeglądanie i sterowanie programem ze zdalnego komputera z zainstalowanym środowiskiem LabVIEW. Zdalna obsługa programów za pośrednictwem protokołu HTTP i przeglądarki internetowej.	2
L14	Łączenie podprogramów ramach projektu. Zapoznanie się z wbudowanymi funkcjami środowiska LabVIEW ułatwiającymi obsługę projektów aplikacji. Tworzenie wykonywalnego pliku samodzielnej aplikacji - Application Builder.	2
L15	Odrabianie zajęć / Wyrównywanie zaległości. Wystawianie ocen.	2
Suma godzin:		30

Metody/Narzędzia dydaktyczne

1	Wykład z prezentacją multimedialną.
2	Dyskusja w trakcie zajęć wykładowych.
3	Laboratorium programistyczne.
4	Praca grupowa

Sposoby oceny

Ocena formująca	
F1	Aktywność w trakcie zajęć
F2	Systematyczność uczęszczania na zajęcia laboratoryjne
Ocena podsumowująca	
P1	Egzamin na ocenę (pytania testowe, wykład)
P2	Średnia ocen z poszczególnych zajęć (oceny z realizacją zleconych zadań, laboratorium)

Obciążenie pracą studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z wykładowcą, w tym:	60
Udział w wykładach	30
Udział w laboratoriach	30
Praca własna studenta, w tym:	
Przygotowanie do laboratoriów – łączna liczba godzin w semestrze	30
Samodzielne przygotowanie do zaliczenia wykładu	35
Łączny czas pracy studenta	125
Sumaryczna liczba punktów ECTS dla przedmiotu	5

Literatura podstawowa i uzupełniająca	
1P	Chruściel M., LabVIEW w praktyce, Wyd. BTC, Legionowo 2008.
2P	Tłaczała W., Środowisko LabVIEW w eksperymencie wspomaganym komputerowo, WNT, W-wa 2002.
3U	National Instruments, Materiały szkoleniowe - LabVIEW Express Basics Interactive Training. CD, National Instruments 2008.
4U	National Instruments, Dokumentacja - G Programming Reference Manual, BridgeVIEW and LabVIEW, National Instruments 2008.

Macierz efektów kształcenia					
Efekt kształcenia	Odniesienie danego efektu kształcenia do efektów zdefiniowanych dla całego programu (PEK)	Cele przedmiotu	Treści programowe	Metody/ Narzędzia dydaktyczne	Sposób oceny
E1	E2A_W05s	C1, C2	W1, W2, W5, W8, L1, L2, L3, L8	1, 2, 3	F1, F2, P1, P2
E2	E2A_W08	C2, C3	W3, W4, W6, W7, L4, L5, L6, L11, L14	1, 2, 3	F1, F2, P1, P2
E3	E2A_U11	C4	W15, L10, L14	1, 3	F1, F2, P1, P2
E4	E2A_U12	C2, C4	W10, W11, W12, L3, L9	1, 2, 3	F1, F2, P1, P2
E5	E2A_U11	C3, C4	W13, W14, L12, L13	1, 3, 4	F1, F2, P1, P2
E6	E2A_K03	C3	W9, W11, L12, L13	1, 3	F1, F2, P1, P2
E7	E2A_K02	C4	W9, W15, L7, L14, L15	1, 3, 4	F1, F2, P1, P2

Formy oceny – szczegóły				
	Na ocenę 2 (ndst)	Na ocenę 3 (dst)	Na ocenę 4 (db)	Na ocenę 5 (bdb)
E1	Student nie rozróżniania typów danych i sposobu oznaczania ich w kodzie źródłowym.	Student rozróżnia typy danych i sposoby wyróżniania ich w kodzie.	Student potrafi wskazać punkty konwersji danych i uzasadnić przyczyny konwersji.	Student potrafi wskazać wymagany typ danych przed rozpoczęciem tworzenia kodu oraz uzasadnić wybór.
E2	Student nie potrafi nazwać symboli struktur kierujących wykonywaniem kodu.	Student rozpoznaje istniejące struktury oraz określa ich funkcjonalność /możliwości.	Student tworzy kod z zastosowaniem struktur na podstawie pisemnej instrukcji oraz zna metodę określania wymagań systemowych aplikacji.	Student potrafi rozpoznać błędnie zastosowane struktury połączona ze wskazaniem poprawnych oraz umie posługiwać się narzędziami do określania wymagań systemowych aplikacji.
E3	Student nie potrafi utworzyć programu na podstawie szczegółowej instrukcji.	Student potrafi utworzyć jednozadaniowy program na podstawie instrukcji.	Student tworzy wielofunkcyjne programy (system) na podstawie sformułowanego problemu.	Student ma umiejętność tworzenia wielofunkcyjnego programu oraz jego kompilacji połączoną ze stworzeniem programu instalacyjnego.
E4	Student nie zna narzędzi do debugowania.	Student wie jak uruchomić narzędzia śledzenia wykonywania kodu. Student zna nazwy popularnych architektur kodu.	Student interpretuje wskazania narzędzi do debugowania programu. Student zna zalety i wady popularnych architektur kodu.	Student płynnie posługuje się narzędziami do analizy kodu oraz posiada umiejętność doboru najlepszej architektury programu.
E5	Student nie zdaje sobie sprawy z możliwości utworzenia zdalnego systemu.	Student potrafi stworzyć system wymieniający dane podstawie instrukcji.	Student potrafi samodzielnie stworzyć prosty system ze zdalnym dostępem do danych.	Student potrafi stworzyć złożony system posiadający możliwość wymiany danych z innym systemem zdalnym oraz program instalacyjny.

E6	Student nie zna żadnej możliwości wymiany danych pomiędzy elementami systemu.	Student zna nazwy funkcji i mechanizmów pozwalających na wymianę danych.	Student potrafi stworzyć system wymieniający dane za pośrednictwem zmiennych globalnych.	Student potrafi stworzyć system wymieniający dane za pośrednictwem zmiennych globalnych i protokołów sieciowych (dftp i http).
E7	Student nie sprawdza wymagań systemowych programów.	Student zna lokalizację i funkcjonalność procedur kontrolnych.	Student posiada umiejętność interpretowania wyników pomiarów zapotrzebowania na pamięć i czas wykonywania kodu.	Student posiada umiejętność doboru odpowiedniej architektury programu połączona z modyfikacją (poprawieniem) kodu w oparciu o wyniki profilowania aplikacji oraz umiejętność tworzenia plików instalacyjnych.

Autor programu:	dr inż. Andrzej Sumorek
Adres e-mail:	a.sumorek@pollub.pl
Jednostka organizacyjna:	Katedra Inżynierii Komputerowej i Elektrycznej