



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Elektryczne i informatyczne układy mechatroniki [N1Eltech1>E-EilUM]

Przedmiot

Kierunek studiów

Elektrotechnika

Rok/Semestr

5/9

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

niestacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

20

Laboratorium

10

Inne

0

Ćwiczenia

0

Projekty/seminaria

10

Liczba punktów ECTS

4,00

Koordynatorzy

dr hab. inż. Cezary Jędrzycka prof. PP
cezary.jedryczka@put.poznan.pl

dr hab. inż. Wojciech Pietrowski
wojciech.pietrowski@put.poznan.pl

Wykładowcy

Wymagania wstępne

Podstawowe wiadomości z zakresu programowania zorientowanego obiektowo, relacyjnych baz danych. Algorytmy i struktury danych. Programowanie w języku wysokiego poziomu, np. C++, Java. Podstawowa wiedza z elektroniki i układów cyfrowych. Zna strukturę i zasadę działania przemiennika częstotliwości. Zna rodzaje, podstawowe charakterystyki i algorytmy sterowania maszyn elektrycznych. Ma świadomość konieczności poszerzania swoich kompetencji, gotowość do podjęcia współpracy w ramach zespołu. Ma świadomość konieczności poszerzania swoich kompetencji, gotowość do podjęcia współpracy w ramach zespołu.

Cel przedmiotu

Zapoznanie z podstawowymi informacjami o platformie zaawansowanego środowiska programowania wizualnego .NET. Nabycie umiejętności projektowania oraz implementowania aplikacji okienkowych w języku C#. Testowanie tworzonych aplikacji. Zapoznanie ze współczesnymi metodami tworzenia oprogramowania dla urządzeń mobilnych pracujących pod kontrolą systemu operacyjnego Android. Poznanie zasady działania omawianych urządzeń oraz algorytmów programów. Zdobycie wiedzy na temat zasady działania, budowy i parametryzacji przekształtnikowych układów napędowych oraz umiejętności praktycznych dotyczących doboru, konfiguracji i uruchamiania aplikacji napędowych stosowanych w przemysłowych układach wykonawczych automatyki.

Przedmiotowe efekty uczenia się

Wiedza:

Ma uporządkowaną wiedzę w zakresie metrologii oraz właściwości i eksploatacji współczesnej aparatury pomiarowej.

Ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie kluczowych zagadnień informatyki niezbędnych dla inżyniera elektryka.

Ma uporządkowaną i podbudowaną teoretycznie wiedzę na temat budowy, zasady działania i eksploatacji transformatorów, maszyn elektrycznych i układów technicznych, zna procesy zachodzących w cyklu ich życia.

Posiada wiedzę dotyczącą budowy, zasady działania, konfiguracji i standardów programowania współczesnych układów napędowych stosowanych w aplikacjach przemysłowych.

Umiejętności:

Potrafi zaprojektować i wykonać, zgodnie z zadaną specyfikacją i przy użyciu właściwych metod, technik, narzędzi i materiałów, typowe układy elektryczne przeznaczone do różnych zastosowań.

Potrafi sformułować algorytm, posługuje się językami programowania oraz narzędziami informatycznymi wykorzystywanymi w inżynierii elektrycznej.

Potrafi posłużyć się właściwie dobranymi narzędziami informatycznymi w celu przeprowadzenia symulacji, projektowania i analizy układów elektrycznych oraz układów napędowych.

Potrafi poprawnie skonfigurować przemiennik częstotliwości do realizacji zadanej aplikacji napędowej.

Potrafi tworzyć i implementować algorytmy sterowania z wykorzystaniem sterowników PLC.

Kompetencje społeczne:

Rozumie znaczenie wiedzy w rozwiązywaniu problemów i podnoszenia kompetencji zawodowych, osobistych i społecznych; ma świadomość, że w technice wiedza i umiejętności bardzo szybko stają się przestarzałe.

Ma świadomość ważności pracy własnej i konieczności przestrzegania zasad etyki zawodowej, jest gotowy do podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania, a także dbałości o dorobek i tradycje zawodu.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wykład:

- Zaliczenie na podstawie kolokwium składającego się z pytań ogólnych i testowych. Skala ocen 0-51% pkt - 2.0, 51-61% pkt. - 3.0, 61-71% pkt - 3.5, 71-81% pkt. – 4.0, 81-91% pkt. – 4.5, 91-100% pkt. – 5.0.

Ćwiczenia laboratoryjne i projektowe:

- sprawdzian i premiowanie wiedzy niezbędnej do realizacji postawionych problemów w danym obszarze zadań laboratoryjnych,
- ocenianie na zajęciach - premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami,
- ocena wiedzy i umiejętności związanych z realizacją zadania ćwiczeniowego, ocena sprawozdania z wykonanego ćwiczenia.

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- proponowanie omówienia dodatkowych aspektów zagadnienia;
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu;
- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium;
- uwagi związane z udoskonaleniem procesu dydaktycznego;

- staranność estetyczną opracowywanych sprawozdań i zadań – w ramach nauki własnej

Treści programowe

Wprowadzenie do platformy .NET. Charakterystyka pakietu Microsoft Visual Studio. Prezentacja środowiska programistycznego. Edycja formularzy. Podstawy programowania w języku C#. Korzystanie z komponentów interfejsu Windows. Biblioteka Windows Forms.

Tematyka zajęć

Wprowadzenie do platformy .NET. Przedstawienie struktury platformy .NET. Rodzaje i typy platformy .NET. Zarys środowiska .NET Framework. Przegląd języków programowania dla platformy .NET. Charakterystyka pakietu Microsoft Visual Studio. Prezentacja środowiska programistycznego. Edycja formularzy. Uruchamianie aplikacji. Tworzenie przykładowej aplikacji dla systemu Windows. Środowisko CLR (ang. Common Language Runtime). Podstawowe funkcje i usługi CLR. Zarządzanie pamięcią i zasobami. Podstawy programowania w języku C#. Omówienie składni języka: instrukcje, zmienne, operatory i typy danych. Dostęp i operacje na plikach. Zasady tworzenia klas, metod, konstruktorów oraz obiektów tych klas. Użycie tablic. Przegląd podstawowych narzędzi zawartych w SDK (ang. Software Development Kit). Zaawansowane elementy języka C#. Dyrektywy preprocesora. Obsługa zdarzeń. Obsługa błędów za pomocą wyjątków. Operacje na łańcuchach znaków. Korzystanie z komponentów interfejsu Windows. Biblioteka Windows Forms. Wykorzystanie przestrzeni nazw, formularz początkowy i jego właściwości. Dodawanie elementów sterujących, określanie ich właściwości i definiowanie funkcji obsługi zdarzeń. Obsługa myszki i klawiatury, menu, pasek statusu, pasek narzędzi. Podstawowe elementy sterujące: przyciski, pola tekstowe, listy rozwijalne itp. Tworzenie wykresów. Okna dialogowe: dialogi standardowe i niestandardowe. Zakładki. Środowisko graficzne GDI+. Pozyskiwanie obiektu graphics. Narzędzia graficzne: czcionki, pióra pędzle. Funkcje rysujące i wypełniające. Przetwarzanie obrazów. Przetwarzanie pikseli, przetwarzanie obrazów.

Tworzenie aplikacji mobilnej w J2ME. Projektowanie interfejsu użytkownika. Grafika dwuwymiarowa i multimedia. Odczytywanie danych i monitorowanie portu COM. Przechowywanie danych w urządzeniach mobilnych, w tym danych geograficznych GIS. Parsowanie danych z odbiornika GPS. Pobieranie i prezentacja danych z bazy danych SQL. Trójwymiarowa grafika OpenGL. Tworzenie aplikacji dostosowanej do różnych urządzeń.

Struktura sterownika PLC, języki programowania i środowiska programistyczne. Struktura i zasada działania przemiennika częstotliwości. Metody i algorytmy sterowania podstawowymi rodzajami maszyn elektrycznych implementowane w współczesnych przemiennikach częstotliwości. Praca grupowa napędów przekształtnikowych w obrębie wspólnej szyny DC. Serwonapędy - historia, stan obecny, struktura blokowa, metody sterowania, bloki funkcyjne PLCopen Motion Control.

Metody dydaktyczne

1. Wykład: prezentacja multimedialna, ilustrowana przykładami podawanymi na tablicy.
2. Ćwiczenia laboratoryjne i projektowe: prezentacja multimedialna prezentacja ilustrowana przykładami podawanymi na tablicy oraz wykonanie zadań podanych przez prowadzącego - ćwiczenia praktyczne. Ćwiczenia laboratoryjne dotyczące parametryzacji i programowania przemysłowych układów napędowych

Literatura

Podstawowa

1. Chappell D.: Zrozumieć platformę .NET, Helion, 2007
2. Duffy J.: .NET Framework 2.0, Zaawansowane programowanie, Helion, 2007
3. Michelsen K.: Język C#. Szkoła programowania, Helion, 2007
4. Griffiths Ian, Adams Matthew, Liberty Jesse, C#. Programowanie, Helion, 2012
5. Andrew Troelsen, Język C# 2010 i platforma .NET 4.0, Wydawnictwo Naukowe PWN, 2011
6. Jacek Matulewski, Dawid Borycki, Grzegorz Krause, Maciej Grabek, Maciej Pakulski, Mateusz Warczak, Jacek Lewan-dowski, Sławomir Orłowski, Visual Studio 2010 dla programistów C#, Helion, 2011
7. Dokumentacja elektroniczna systemu programowania Visual Studio.NET
8. Hello, Andriod. Programowanie na platformę Google dla urządzeń mobilnych. Helion, 2011.
9. T. Imieliński, H. F. Korth, Mobile Computing. KLUWER, 1996.
10. J. Januszewski, Systemy satelitarne GPS, Galileo i inne, PWN, 2010.
11. S. Shekhar, S. Chawla, Spatial database: A Tour. Prentice Hall, 2003.

Uzupełniająca

1. Burton K.: .NET CLR. Księga eksperta, Helion, 2002
2. Liberty J.: C# 2005. Wprowadzenie, O'Reilly, 2007
3. Hejlsberg Anders, Torgersen Mads, Wiltamuth Scott, Golde Peter, Język C#. Programowanie, Helion, 2010
4. Mike Snell, Lars Powers, Microsoft Visual Studio 2010. Księga eksperta, Helion 2011
5. M. P. Clark, Wireless Access Networks: Fixed Wireless Access and WLL Networks Design and Operation, Wiley, 2000.
6. W. Hołubowicz, P. Płóciennik, Cyfrowe Systemy Telefonii Komórkowej GSM 900, GSM 1800, UMTS, 1998.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	55	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	45	2,00