

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Implementacja komputerowych architektur sprzętowych		Kod 1010531161010557595
Kierunek studiów Automatyka i robotyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 3 / 6
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obieralny
Stopień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 15 Ćwiczenia: - Laboratoria: - Projekty/seminaria: 30		Liczba punktów 3
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (ogólnouczelniany, z innego kierunku) kierunkowy z danego kierunku		
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki		Podział ECTS (liczba i %)
Odpowiedzialny za przedmiot / wykładowca:		
dr inż. Janusz Pochmara email: Janusz.Pochmara@put.poznan.pl tel. 61 6652184 Informatyki ul. Piotrowo 3A, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Słuchacz powinien dysponować podstawami z fizyki w zakresie elektryczności, przepływu prądu, a znajomość podstaw programowania pozwoli w łatwy sposób poznać ideę programowania drabinkowego.
2	Umiejętności:	Powinien również posiadać umiejętność rozwiązywania podstawowych problemów w zakresie podstaw automatyki oraz technik sterowania. Umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien również rozumieć konieczność poszerzania swoich kompetencji / mieć gotowość do podjęcia współpracy w ramach zespołu.
3	Kompetencje społeczne	Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
Cel przedmiotu:		
1. Student pozna w jaki sposób tworzy się podstawowe architektury komputerowe. 2. Potrafi analizować zdiasemblowane proste programy ? określa ich funkcjonalność 3. Umie dokonać opisu i symulacji cyfrowych układów kombinacyjnych i sekwencyjnych. 4. Umie zaprojektować i wykonać podstawowy system mikroprocesorowy 5. Potrafi tworzyć oprogramowanie na poziomie sprzętowym. 6. Ma wiedzę na temat tego jak działa mikroprocesor ze sterowaniem sprzętowym 7. Kształtowanie u studentów umiejętności pracy zespołowej. 8. Będzie potrafił w sposób intuicyjny wykorzystywać interfejs graficzny w procesach sterowania. 9. Zapozna studentów z narzędziami dzięki, którym łatwiejsze staje się budowanie interaktywnych aplikacji komputerowych. 10. Student zapozna się z technikami rozproszonego programowania, które są łatwo dostępne z poziomu języka Java. Ze strony sprzętowej pozna podstawowe struktury sieci przemysłowych oraz sposoby akwizycji danych przy pomocy sterowników przemysłowych.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
1. ma uporządkowaną wiedzę w zakresie architektury komputerów, systemów i sieci komputerowych oraz systemów operacyjnych w tym systemów operacyjnych czasu rzeczywistego; - [K_W9] 2. Zna i rozumie budowę i zasady działania programowalnych sterowników przemysłowych a także ich analogowych i cyfrowych układów peryferyjnych; zna i rozumie zasadę działania podstawowych interfejsów komunikacyjnych stosowanych w przemysłowych systemach sterowania; - [K_W19]		
Umiejętności:		

1. potrafi korzystać z wybranych narzędzi szybkiego prototypowania układów automatyki i robotyki; - [K_U13]
2. potrafi dobrać parametry i nastawy podstawowego regulatora przemysłowego oraz skonfigurować i zaprogramować przemysłowy sterownik programowalny; - [K_U18]
3. potrafi dokonać identyfikacji i sformułować specyfikację prostych zadań inżynierskich z zakresu automatyki i robotyki; - [K_U23]
4. potrafi zaprojektować i zrealizować lokalną sieć teleinformatyczną (w tym przemysłową) przez dobór i konfigurację elementów i urządzeń komunikacyjnych (przewodowych i bezprzewodowych); - [K_U28]
Kompetencje społeczne:
1. posiada świadomość odpowiedzialności za pracę własną oraz gotowość podporządkowania się zasadom pracy w zespole i ponoszenia odpowiedzialności za wspólnie realizowane zadania; potrafi kierować małym zespołem, wyznaczać cele i określać priorytety prowadzące do realizacji zadania; - [K_K3]

Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

a) w zakresie wykładów:

na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,

b) w zakresie laboratoriów:

na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:

i. ocenę wiedzy i umiejętności wykazanych na kolokwium w formie testu otwartego, zawierającego 20 pytań z listy 100 zagadnień, które zostaną udostępnione studentom. Obowiązują punktowe kryteria oceniania postaci:

<65pkt ndst,

65-74pkt dst,

65-75 dst+,

76-84 db,

85-94 db+

>95 bdb)

ii. omówienie wyników kolokwium,

b) w zakresie zajęć praktycznych - projektów weryfikowanie założonych efektów kształcenia realizowane jest przez:

i. ocenę przygotowania studenta do poszczególnych sesji zajęć w postaci wykonywania projektów podczas spotkań na laboratoriach projektowych (sprawdzian wejściowy) oraz ocenę umiejętności związanych z realizacją poszczególnych ćwiczeń,

100 pkt, w tym:

15x4pkt sprawozdania,

2x20pkt sprawdziany

ii. obowiązują punktowe kryteria oceniania postaci

<65pkt ndst,

65-74pkt dst,

65-75 dst+,

76-84 db,

85-94 db+,

>95 bdb

iii. istnieje możliwość zdobycia premii punktów za rozwiązania autorskie (niepowtarzalne), dla przedstawianych problemów podczas ćwiczeń laboratoryjnych

Treści programowe

Program wykładu obejmuje następujące zagadnienia (15 wykładów):

1. Cyfrowy komputer ? paradygmaty, modele architektury
2. Systemy liczbowe ? reprezentacja danych w pamięci, obsługa debugera
3. Inżynieria rewersyjna ? analizowanie programów na podstawie kodu wykonywalnego
4. Postulaty von Neumanna, Maszyna Turinga, Sieci Petriego
5. Architektura mikroprocesora na bazie intelowskiej - standardy
6. Klasyfikacja układów logicznych - kombinacyjne oraz sekwencyjne
7. Układy arytmetyczne, automaty
8. Zarządzanie pamięcią
9. Organizacja procesora, systemy wieloprocesorowe

<p>10. Asembler procesorów intelowskich</p> <p>11. Programowanie niskopoziomowe ? model funkcjonalny</p> <p>12. Programowanie niskopoziomowe ? makra</p> <p>13. Organizacja warstwowa</p> <p>14. Przykłady rozwiązań technicznych oraz perspektywy rozwojowe ? przedstawienie standardów stosowanych w nowoczesnych aplikacjach przemysłowych</p> <p>15. Systemy wbudowane organizacja oraz praktyczne przykłady</p> <p>Obejmuje implementacje podstawowych układów w oparciu o:</p> <p>1. Wprowadzenie do środowiska Visual Studio C++</p> <p>2. Organizacja pamięci w typowych komputerach klasy PC, mapa pamięci</p> <p>3. Ingerencje użytkownika w kod wykonywalnych programów</p> <p>4. Projektowanie prostych układów ALU - kalkulator</p> <p>5. Maszyna Turinga ? program symulacyjny</p> <p>6. Projektowanie systemu mikroprocesorowego na bazie architektury intelowskiej ze sterowaniem sprzętowym</p> <p>7. Projektowanie systemu mikroprocesorowego na bazie architektury intelowskiej ze sterowaniem programowalnym</p> <p>8. Programowalne układy wejścia/wyjścia</p> <p>9. Architektura kart graficznych</p> <p>10. Aplikacje niskopoziomowe ? obsługa karty graficznej</p> <p>11. Przetwarzanie równoległe CUDA</p> <p>Do ćwiczeń praktycznych została wybrana architektura intelowska ze względu na swą popularność i dostępność w większości produkowanych procesorów używanych w komputerach klasy PC.</p> <p>Metody dydaktyczne:</p> <p>1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, demonstracja, pogadanka, spotkanie z praktykiem, wycieczka do centrum monitoringu i sterowania CW PP.</p> <p>2. ćwiczenia laboratoryjne: rozwiązywanie zadań, wykonywanie eksperymentów, dyskusja, praca w zespole, demonstracja</p> <p>3. wykład gościnny: prezentacja multimedialna zaproszonego inżyniera praktyka z przemysłu (np.: firma Siemens, PhoenixContact, Politechnika Poznańska)</p>		
<p>Literatura podstawowa:</p> <p>1. Programowanie w języku Assembler : poradnik zawodowy / Andrzej Jordan ; Ośrodek Badawczo-Rozwojowy Informatyki. Autor: Jordan, Andrzej (1936-). Wydawnictwa Szkoleniowe OBRl, 1973.</p> <p>2. Systemy komputerowe automatyki przemysłowej. T. 1, Sprzęt i oprogramowanie / Antoni Niederliński. Autor: Niederliński, Antoni. Wydawnictwa Naukowo-Techniczne, 1985.</p>		
<p>Literatura uzupełniająca:</p> <p>1. Visual Studio 2012 i .NET 4.5 : poradnik dla zaawansowanych programistów : ponad 40 przepisów skutecznego łączenia wydajnych możliwości .NET 4.5 i Visual Studio 2012 / Abhishek Sur ; przekł.: Jakub Niedźwiedz. Autor: Sur, Abhishek. APN Promise, 2013.</p>		
<p>Bilans nakładu pracy przeciętnego studenta</p>		
<p>Czynność</p>		<p>Czas (godz.)</p>
<p>1. udział w zajęciach projektowych:</p> <p>2. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu</p> <p>3. napisanie programu / programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)</p> <p>4. projekty</p> <p>5. przygotowanie do sprawdzianów</p> <p>6. udział w wykładach</p> <p>7. omówienie projektów</p> <p>8. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym</p>		<p>30</p> <p>2</p> <p>8</p> <p>5</p> <p>10</p> <p>15</p> <p>2</p> <p>3</p>
<p>Obciążenie pracą studenta</p>		
<p>forma aktywności</p>	<p>godzin</p>	<p>ECTS</p>
Łączny nakład pracy	75	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	47	2
Zajęcia o charakterze praktycznym	45	1