KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu
Architektura systemów komputerowych

Przedmiot

Kierunek studiów
Informatyka

Rok/semestr
2/4

Studia w zakresie (specjalność)

Profil studiów
ogólnoakademicki

Poziom studiów
pierwszego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład	Laboratoria	Inne (np. online)
30	30	
Ćwiczenia	Projekty/seminaria	

Liczba punktów ECTS

5

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:
dr hab. inż. Piotr Zielniewicz
email: Piotr.Zielniewicz@cs.put.poznan.pl
tel: (48)(61) 665-2935
wydział: Instytut Informatyki
adres: 60-965 Poznań, ul. ul. Piotrowo 2

Odpowiedzialny za przedmiot/wykładowca:
dr inż. Rafał Klaus
email: Rafal.Klaus@cs.put.poznan.pl
tel: (48)(61) 665-2574
wydział: Instytut Informatyki
adres: 60-965 Poznań, ul. ul. Piotrowo 2

Wymagania wstępne

Student powinien posiadać podstawową wiedzę z matematyki dyskretnej (pozycyjne systemy liczbowe, konwersja systemów liczbowych, logiki formalne, algebra Boola), techniki cyfrowej i programowania niskopoziomowego .
Student powinien posiadać umiejętność rozwiązywania podstawowych problemów dotyczących: projektowania układów elektronicznych sterowania silnikami (DC i krokowymi), układów elektronicznych czujników optycznych, zbliżeniowych i innych, posługiwania się współczesnymi systemami operacyjnymi oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
Powinien również rozumieć konieczność poszerzania swoich kompetencji oraz mieć gotowość do podjęcia współpracy w ramach zespołu. Ponadto student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.

Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy dotyczącej budowy i zasady działania systemów komputerowych, w zakresie: architektur i struktur systemów mikroprocesorowych, architektury mikroprocesorów, mikrokontrolerów, układów pamięciowych, interfejsów komunikacyjnych, programowalnych układów wejścia-wyjścia, systemów przerwań, układów DMA, pamięci podręcznych, magistral systemowych, systemów wbudowanych, paralelizmu na poziomie architektury, efektywności systemów komputerowych oraz zasad programowania niskopoziomowego.
2. Rozwijanie u studentów umiejętności rozwiązywania problemów optymalnego programowania niskopoziomowego systemów mikroprocesorowych, programowania interfejsów i chipsetów, projektowania, budowy i uruchamiania systemów mikroprocesorowych, budowy prostych robotów z systemem mikroprocesorowym, tworzenia dokumentacji projektowej, powykonawczej i techniczno-rozruchowej.
3. Kształtowanie u studentów umiejętności pracy zespołowej i twórczego kreatywnego myślenia poprzez zastosowanie autorskiego systemu szkolenia (Academy of Creative Action).

Przedmiotowe efekty uczenia się

Wiedza

1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów komputerowych i systemów wbudowanych oraz sprzętowego wsparcia systemów operacyjnych.
2. ma uporządkowaną i podbudowaną teoretycznie wiedzę szczegółową z zakresu niskopoziomowych języków programowania, sprzętowych interfejsów komunikacji człowiek-komputer.
3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w zakresie architektur systemów komputerowych, efektywności systemów komputerowych, zasad programowania niskopoziomowego, problemów komputerów biologicznych, optycznych i kwantowych.
4. ma podstawową wiedzę o cyklu życia systemów informatycznych sprzętowych i programowych w zakresie niezbędnym do realizacji zadań warsztatowo-laboratoryjnych.
5. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu prostych zadań informatycznych z zakresu budowy systemów komputerowych, systemów wbudowanych, interfejsów komunikacji człowiek-komputer, inżynierii oprogramowania w zakresie niezbędnym do realizacji zadań warsztatowo-laboratoryjnych.

Umiejętności

1. potrafi planować i przeprowadzać eksperymenty, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski w ramach działań warsztatowo-laboratoryjnych.
2. potrafi zastosować odpowiednio dobrane metody do sformułowania i rozwiązywania zadań w trakcie zajęć warsztatowo-laboratoryjnych (np. podczas projektowania systemu mikroprocesorowego z elementami mechatronik).
3. potrafi wybrać język programowania odpowiedni do rozwiązania danego problemu podczas realizacji cyklów warsztatowo-laboratoryjnych oraz potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi.
4. potrafi zaprojektować, zbudować i oprogramować proste systemy mikroprocesorowe i wbudowane.
5. potrafi organizować, współdziałać i pracować w grupie podczas projektowania systemu mikroprocesorowego.

Kompetencje społeczne

1. zna przykłady i rozumie przyczyny wadliwie działających systemów, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia w zakresie budowy i oprogramowania mikroprocesorowych systemów sterowania
2. potrafi myśleć i działać w sposób przedsiębiorczy uwzględniając korzyści biznesowe oraz uwarunkowania społeczne.
3. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe
4. prawidłowo identyfikuje i rozstrzyga dylematy związane z wykonywaniem zawodu - podczas prac zespołowych warsztatowych analiza lojalności wobec jednostek w grupie a powierzonego zadania
5. ma świadomość roli społecznej absolwenta uczelni technicznej, a zwłaszcza rozumie potrzebę formułowania i przekazywania społeczeństwu, w szczególności poprzez środki masowego przekazu, informacji i opinii dotyczących osiągnięć techniki i innych aspektów działalności inżynierskiej - zaangażowanie w organizację zawodów robotów z działaniami szkoleniowymi i promocyjnymi

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza nabyta w ramach wykładu jest weryfikowana przez egzamin, na który składa się 8-12 pytań testowych pojedynczego wyboru, 4-8 pytań testowych wielokrotnego wyboru oraz 3-4 pytań/zadań o charakterze otwartym różnie punktowanych. Próg zaliczeniowy: 50% punktów.
Umiejętności nabyte w ramach zajęć warsztatowo-laboratoryjnych weryfikowane są na podstawie zaliczania wykonanego urządzenia z systemem mikroprocesorowym i elementami mechatroniki. Ocenie podlega skala trudności budowy systemu mikroprocesorowego, kreatywność wykorzystania mechatroniki, jakość wykonania systemu, terminowość realizacji. Ponadto mogą być realizowane trzy sprawdziany formujące z zakresu konstrukcji i programowania w asemblerze ADuC842, arduino, rospberry. Próg zaliczeniowy 50% punktów.

Treści programowe

Wykłady:
Wprowadzenie do architektury systemów komputerowych
Struktura systemu komputerowego
Układy programowalne transmisji i generator interwałów czasowych
Układy programowalne systemu przerwań i DMA
Układy pamięciowe
Interfejsy komunikacyjne
Architektura procesora
Pamięć podręczna
Model programowy procesora
Magistrale systemowe
Rodziny popularnych mikrokontrolerów w tym z jądrem 8051, mikrokontroler AduC842, systemy arduino i rospberry pi, programowanie w asemblerze, C oraz Pythton.

Laboratoria:

- część laboratoryjna: narzędzia uruchomieniowe mikrokontrolerów, poznają środowiska programowe i narzędzia sprzętowe oraz ćwiczą metody uruchamiania, inspekcji kodu programu i wyszukiwania błędów w konstrukcji sprzętowych. Poznają zasady obsługi wyświetlaczy siedmiosegmentowych, LSD, sterownia silnikami, obsługi czujników i innych układów wykonawczych, programowania zasobów sprzętowych mikrokontrolerów.
- część warsztatowa: w pracy zespołowej studenci projektują, wykonują i oprogramowują autonomicznego mikroprocesorowego robota. Poznają zasady budowy dokumentacji powykonawczej i DTR. Muszą obronić urządzenie, oprogramowanie i dokumentację. W pracy zespołowej lub indywidualnej przygotowują szkolenia dydaktyczne oraz zawody swoich robotów dla szkół średnich (wydarzenie RoboDay)
Warsztaty realizowane w ramach programu autorskiego nauki kreatywności i twórczego myślenia metodami design thinking, learning by doing, project based learning.

Metody dydaktyczne

1. Wykład: slajdy, prezentacja multimedialna, prezentacja ilustrowana przykładami, dyskusja z wykorzystaniem tablicy, rozwiązywanie zadań sprzętowo-programowych, pokaz multimedialny w postaci filmów np. z budowy robotów, demonstracja robotów zrealizowanych w poprzednich latach.
2. Ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia problemowe, wykonywanie eksperymentów pomiarowych, dyskusja z badaniami on-line na analizowanych systemach mikroprocesorowych, praca indywidualna i w zespołach, pokaz multimedialny z zawodów poprzednich lat z analizą błędów, warsztaty jako kluczowy elementem nauki kreatywności twórczej, studium przypadków podczas badania konkretnych systemów, demonstracja przykładowych zagadnień.

Literatura

Podstawowa

1. Organizacja i architektura systemu komputerowego, W. Stallings, WNT, Warszawa, 2004
2. Struktura organizacyjna i architektura systemów komputerowych, L. Null, J. Lobur, Helion, Gliwice, 2004
3. Anatomia PC, P. Metzger, Helion, Gliwice, 2007

Uzupełniająca

1. Architektura komputerów, J. Biernat, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2005
2. Computer Organization and Design, D. Patterson, J. Hennessy, Morgan Kaufmann, 2008
3. Klaus R., Szymaniak P.: “Prototypowanie 3D robota pirotechnicznego”, Mechanika z.103 nr 351/2014,Zeszyty Naukowe Politechnika Opolska Opole 2014; ISBN 978-83-64056-49-9
4. Klaus R. Agilecoach na Wydziale Informatyki Politechniki Poznańskiej, http://biuletyn.pti.org.pl/BiuletynPTI_2016-04.pdf
5. Klaus R. RoboDay na Wydziale Informatyki Politechniki Poznańskiej, http://biuletyn.pti.org.pl/BiuletynPTI-2016-03.pdf

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	125	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	64	3.0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwiów/egzaminu, wykonanie projektu)[1]	61	2.0

[1] niepotrzebne skreślić lub dopisać inne czynności