



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Architektura systemów komputerowych [N1Inf1>ASK]

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
niestacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
16

Laboratorium
24

Inne (np. online)
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

5,00

Koordynatorzy

dr inż. Rafał Klaus
rafal.klaus@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student powinien posiadać podstawową wiedzę, umiejętności oraz kompetencje społeczne w zakresie: matematyki: pozycyjne systemy liczbowe, konwersji systemów liczbowych, logiki formalnej, algebry Boola, elektroniki: elektrotechnika, elementy półprzewodnikowe, układy tranzystorowe, wzmacniacze, komparatory, filtry, modulatory, generatory, przetworniki a/c i c/a), techniki cyfrowej, programowania niskopoziomowego, systemów operacyjnych, umiejętność pozyskiwania informacji ze wskazanych źródeł, rozumieć konieczność poszerzania swoich kompetencji, 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 pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie - podczas projektowania systemu mikroprocesorowego
2. potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych podczas pracy zespołowej w ramach działań warsztatowo-laboratoryjnych
3. potrafi planować i przeprowadzać eksperymenty, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski w ramach działań warsztatowo-laboratoryjnych
4. potrafi zastosować odpowiednio dobrane metody do sformułowania i rozwiązywania zadań w trakcie zajęć warsztatowo-laboratoryjnych (np. podczas projektowania systemu mikroprocesorowego z elementami mechatroniki)
5. potrafi efektywnie uczestniczyć w inspekcji oprogramowania niskopoziomowego oraz ocenić architekturę oprogramowania niskopoziomowego z punktu widzenia wymagań pozafunkcyjnych
6. potrafi wybrać język programowania odpowiedni do rozwiązania danego problemu podczas realizacji cykli warsztatowo-laboratoryjnych oraz potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi
7. ma umiejętność formułowania algorytmów i ich programowania z użyciem przynajmniej jednego z popularnych narzędzi podczas realizacji cykli warsztatowo-laboratoryjnych
8. potrafi zaprojektować, zbudować i oprogramować proste systemy mikroprocesorowe i wbudowane
9. potrafi organizować, współdziałać i pracować w grupie podczas projektowania systemu mikroprocesorowego

Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe
2. 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
3. potrafi myśleć i działać w sposób przedsiębiorczy uwzględniając korzyści biznesowe oraz

uwarunkowania społeczne

4. 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 szkoleń i promocji

5. prawidłowo identyfikuje i rozstrzyga dylematy związane z wykonywaniem zawodu - podczas prac zespołowych warsztatowych analiza lojalności wobec jednostek w grupie a powierzonego zadania

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się 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 na każdym zajęciach oraz sprawdzianów etapowych.

c) w zakresie warsztatów: na podstawie oceny z zaliczenia wykonanego urządzenia

Ocena podsumowująca:

W części wykładowej: ocena pracy kontrolnej oraz ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym. Egzamin trwa 60 min, składa się 3 pytań testowych wielokrotnego wyboru oraz 3 pytań/zadań otwartych. Aby zaliczyć egzamin należy uzyskać co najmniej 51% maksymalnej liczby punktów w tym z pracy kontrolnej.

Umiejętności nabyte w ramach zajęć warsztatowo-laboratoryjnych weryfikowane są na podstawie zaliczenia 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 assemblerze ADuC842, arduino, raspberry. Próg zaliczeniowy 50% punktów.

Możliwe jest uzyskanie dodatkowych ocen za aktywność podczas zajęć w szczególności za: realizację indywidualnych prac problemowych związanych z przedmiotem np. udoskonalenie materiałów i pomocy dydaktycznych, realizację indywidualnych prac tematycznych związanych z organizacją szkoleń.

Treści programowe

W ramach kolejnych wykładów studenci zdobywają wiedzę z zakresu: 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 raspberry pi, programowanie w assemblerze, C oraz Python. Realizacja pracy kontrolnej.

W ramach ćwiczeń laboratoryjnych studenci poznają: 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. W część warsztatowa: w pracy zespołowej studenci projektują, wykonują i oprogramowują system mikroprocesorowy.

Tematyka zajęć

brak

Metody dydaktyczne

1. wykład: slajdy, prezentacja multimedialna, prezentacja ilustrowana przykładami, dyskusja z wykorzystaniem tablicy, rozwiązywanie zadań sprzętowo-programowych, pokaz multimedialny w postaci filmów, studium przypadków podczas badania konkretnych systemów, demonstracja przykładowych zagadnień. Realizacja pracy kontrolnej.

2. ćwiczenia laboratoryjne: rozwiązywanie zadań, ćwiczenia problemowe, wykonywanie eksperymentów pomiarowych, dyskusja z badaniami on-line na analizowanych systemach mikroprocesorowych, praca indywidualna i w zespołach, pokazy multimedialne.

Literatura

Podstawowa

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

Uzupełniająca

1. J. Biernat, Architektura komputerów, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2005
2. D. Patterson, J. Hennessy, Computer Organization and Design, Morgan Kaufmann, 2008
3. Klaus R., Szymaniak P.: Prototypowanie 3D robota pirotechnicznego, Mechanika z.103 nr 351/2014, Zeszyty Naukowe Politechniki 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	5,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	42	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	83	3,00