

KARTA OPISU MODUŁU KSZTAŁCENIA			
Nazwa modułu/przedmiotu Architektura systemów komputerowych			Kod 1010511341010511920
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki		Rok / Semestr 2 / 4
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny	
Stopień studiów: I stopień		Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: 30 Projekty/seminaria: -			Liczba punktów 5
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) ogólnouczelniany	
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne nauki techniczne			Podział ECTS (liczba i %) 5 100% 5 100%
Odpowiedzialny za przedmiot / wykładowca: Odpowiedzialny za przedmiot / wykładowca: <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> dr inż. Piotr Zielniewicz email: Piotr.Zielniewicz@cs.put.poznan.pl tel. (48)(61) 665-2935 Instytut Informatyki 60-965 Poznań, ul. ul. Piotrowo 2 </div> <div style="width: 48%;"> dr inż. Rafał Klaus email: Rafal.Klaus@cs.put.poznan.pl tel. (48)(61) 665-2574 Instytut Informatyki 60-965 Poznań, ul. ul. Piotrowo 2 </div> </div>			
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:			
1	Wiedza:	Student powinien posiadać podstawową wiedzę z: - 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	
2	Umiejętności:	Student powinien posiadać umiejętność rozwiązywania podstawowych problemów dotyczących: projektowania układów elektronicznych sterowania silnikami (DC i krokowymi), układów elektronicznych czujników optycznych, zbliżeniowych i innych, posługiwanie się współczesnymi systemami operacyjnymi oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.	
3	Kompetencje społeczne	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.			
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia			
Wiedza:			

<p>1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie architektury systemów komputerowych i systemów wbudowanych oraz sprzętowego wsparcia systemów operacyjnych - [K1st_W3]</p> <p>2. ma uporządkowaną i podbudowaną teoretycznie wiedzę szczegółową z zakresu niskopoziomych języków programowania, sprzętowych interfejsów komunikacji człowiek-komputer - [K1st_W4]</p> <p>3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych w zakresie: efektywności systemów komputerowych, zasad programowania niskopoziomowego, problemów komputerów biologicznych, optycznych i kwantowych. - [K1st_W5]</p> <p>4. ma podstawową wiedzę o cyklu życia systemów informatycznych sprzętowych i programowych w zakresie niezbędnym do realizacji zadań warsztatowo-laboratoryjnych - [K1st_W6]</p> <p>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 - [K1st_W7]</p>
<p>Umiejętności:</p> <p>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 oraz robota klasy line follower - [K1st_U1]</p> <p>2. potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji przedsięwzięć informatycznych podczas pracy zespołowej w ramach działań warsztatowo-laboratoryjnych - [K1st_U2]</p> <p>3. potrafi planować i przeprowadzać eksperymenty, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski w ramach działań warsztatowo-laboratoryjnych - [K1st_U3]</p> <p>4. potrafi zastosować odpowiednio dobrane metody do sformułowania i rozwiązywania zadań w trakcie zajęć warsztatowo-laboratoryjnych (np. podczas projektowania systemu mikroprocesorowego oraz robota klasy line follower) - [K1st_U4]</p> <p>5. potrafi efektywnie uczestniczyć w inspekcji oprogramowania niskopoziomowego oraz ocenić architekturę oprogramowania niskopoziomowego z punktu widzenia wymagań pozafunkcyjnych - [K1st_U9]</p> <p>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 - [K1st_U10]</p> <p>7. ma umiejętność formułowania algorytmów i ich programowania z użyciem przynajmniej jednego z popularnych narzędzi podczas realizacji cykli warsztatowo-laboratoryjnych - [K1st_U11]</p> <p>8. potrafi zaprojektować, zbudować i oprogramować proste systemy mikroprocesorowe i wbudowane - [K1st_U13]</p> <p>9. potrafi organizować, współdziałać i pracować w grupie podczas projektowania systemu mikroprocesorowego oraz robota klasy line follower - [K1st_U18]</p>
<p>Kompetencje społeczne:</p> <p>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K1st_K1]</p> <p>2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia w zakresie budowy i oprogramowania mikroprocesorowych systemów sterowania - [K1st_K2]</p> <p>3. potrafi myśleć i działać w sposób przedsiębiorczy uwzględniając korzyści biznesowe oraz uwarunkowania społeczne - [K1st_K3]</p> <p>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 zawodów robotów z działaniami szkoleniowymi i promocyjnymi - [K1st_K4]</p> <p>5. prawidłowo identyfikuje i rozstrzyga dylematy związane z wykonywaniem zawodu - podczas prac zespołowych warsztatowych analiza lojalności wobec jednostek w grupie a powierzonego zadania - [K1st_K5]</p>

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 wejściówek na każdym zajęciach.

c) w zakresie warsztatów:

- na podstawie oceny zaliczenia wykonanego urządzenia i jego prezentacji podczas RoboDay.

Ocena podsumowująca:

W części laboratoryjnej z każdej z 10 wejściówek student uzyskuje od 2 do 5 punktów: 46-50=db., 7-45=dst.plus, 25-36=dst., 0-24=ndst.

Jeśli podejmie decyzję uczestniczenia w części warsztatowej może uzyskać max 5 punktów: 54-55=b.db., 51-53=db plus., 46-50=db., 37-45=dst.plus, 25-36=dst., 0-24=ndst

- W części warsztatowej dostaje punkty (do 5) za zaprojektowanie i zbudowanie przez studenta urządzenia mikroprocesorowego, jego oprogramowania i dokumentacji z realizacji zajęć warsztatowych,

- ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym

- egzamin trwa 60 min, składa się z 8-12 pytań testowych pojedynczego wyboru, 4-8 pytań testowych wielokrotnego wyboru oraz 3-4 pytań/zadań otwartych. Za w pełni prawidłowe odpowiedzi na pytania testowe można uzyskać 60% punktów, za pytania/zadania otwarte - 40% punktów. Aby zaliczyć egzamin należy uzyskać co najmniej 51% punktów.

Możliwe jest uzyskanie dodatkowych ocen za aktywność podczas zajęć laboratoryjnych, a szczególnie za:

- realizację indywidualnych prac problemowych związanych z przedmiotem na rzecz laboratorium np.

udoskonalenie materiałów i pomocy dydaktycznych,

- realizację indywidualnych prac tematycznych związanych z organizacją szkoleń i zawodów

Treści programowe

W ramach kolejnych wykładów studenci zdobywają wiedzę dotyczącą:

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

Rodzina mikrokontrolerów z jądrem 8051

Mikrokontroler AduC842 i inne aspekty warsztatów

W ramach ćwiczeń laboratoryjnych studenci poznają:

- 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 (fakultatywna): 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:		
<ol style="list-style-type: none"> 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. Computer Organization and Design, D. Patterson, J. Hennessy, Morgan Kaufmann, 2008 4. Anatomia PC, P. Metzger, Helion, Gliwice, 2007 5. Organizacja i architektura systemu komputerowego, W. Stallings, WNT, Warszawa, 2004 6. Struktura organizacyjna i architektura systemów komputerowych, L. Null, J. Lobur, Helion, Gliwice, 2004 7. Anatomia PC, P. Metzger, Helion, Gliwice, 2007 		
Literatura uzupełniająca:		
<ol style="list-style-type: none"> 1. Architektura komputerów, J. Biernat, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2005 2. Architektura komputerów, J. Biernat, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2005 3. Computer Organization and Design, D. Patterson, J. Hennessy, Morgan Kaufmann, 2008 4. 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 5. Klaus R. Agilecoach na Wydziale Informatyki Politechniki Poznańskiej, http://biuletyn.pti.org.pl/BiuletynPTI_2016-04.pdf 6. Klaus R. RoboDay na Wydziale Informatyki Politechniki Poznańskiej, http://biuletyn.pti.org.pl/BiuletynPTI-2016-03.pdf 		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. udział w zajęciach laboratoryjnych / ćwiczeniach:	30	
2. przygotowanie do ćwiczeń laboratoryjnych:	15	
3. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności realizacją praktyczną ćwiczeń laboratoryjnych	2	
4. projektowanie, wykonanie i napisanie programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)	15	
5. przygotowanie do sprawdzianów	5	
6. udział w wykładach	30	
7. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron	10	
8. przygotowanie do egzaminu i obecność na egzaminie: 16 godz. + 2 godz.	18	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	125	5
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	64	3
Zajęcia o charakterze praktycznym	60	3