

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Systemy czasu rzeczywistego</b>		Kod <b>1010531161010550353</b>
Kierunek studiów <b>Automatyka i robotyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>3 / 6</b>
Ścieżka obieralności/specjalność <b>-</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obieralny</b>
Stopień studiów: <b>I stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>15</b> Ćwiczenia: <b>-</b> Laboratoria: <b>30</b> Projekty/seminaria: <b>-</b>		Liczba punktów <b>3</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (ogólnouczelniany, z innego kierunku) <b>kierunkowy z danego kierunku</b>		
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>3 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b>		
dr inż. Jarosław Majchrzak email: Jaroslaw.Majchrzak@put.poznan.pl tel. 61 6652847 Wydział Informatyki ul. Piotrowo 3 60-965 Poznań		mgr inż. Michał Kowalski email: Michal.Kowalski@put.poznan.pl tel. 61 6652100 Wydział Informatyki ul. Piotrowo 3 60-965 Poznań
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z programowania w językach wysokiego poziomu (wymagane programowanie strukturalne w C, obiektowe C++, wskazane programowanie w językach LAD, STL ), tworzenia algorytmu, posługiwania się narzędziami do programowania
2	<b>Umiejętności:</b>	Powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu objętego wymaganą wiedzą oraz umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien również rozumieć konieczność poszerzania swoich kompetencji.
3	<b>Kompetencje społeczne</b>	Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
<b>Cel przedmiotu:</b>		
1. Przedstawienie zagadnień związanych z systemami czasu rzeczywistego stosowanymi w procesach sterowania. 2. Analiza teoretycznych oraz praktycznych problemów z związanych z projektowaniem, działaniem i testowaniem programów w systemach czasu rzeczywistego. 3. Wykształcenie umiejętności praktycznej implementacji oraz oceny cech działania wybranych systemów sterowania czasu rzeczywistego		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. ma uporządkowaną wiedzę w zakresie architektur komputerów, systemów i sieci komputerowych oraz systemów operacyjnych w tym systemów operacyjnych czasu rzeczywistego; - [K_W9] 2. ma podstawową wiedzę w zakresie architektur i programowania systemów mikroprocesorowych, zna wybrane języki wysokiego i niskiego poziomu programowania mikroprocesorów, - [K_W13] 3. zna i rozumie budowę i zasady działania programowalnych sterowników przemysłowych a także ich analogowych i cyfrowych układów peryferyjnych; zna i rozumie zasadę działania; - [K_W19]		
<b>Umiejętności:</b>		
1. potrafi skonstruować algorytm dla prostego zadania inżynierskiego oraz zaimplementować, przetestować i uruchomić go w wybranym środowisku programistycznym na komputerze; - [K_U26] 2. potrafi skonstruować algorytm rozwiązania prostego zadania pomiarowego i obliczeniowo-sterującego oraz zaimplementować, przetestować i uruchomić go w wybranym środowisku - [K_U27] 3. potrafi zaprojektować i zrealizować lokalną sieć teleinformatyczną (w tym przemysłową) przez dobór i konfigurację elementów i urządzeń komunikacyjnych (przewodowych i bezprzewodowych); - [K_U28]		
<b>Kompetencje społeczne:</b>		
1. posiada świadomość konieczności profesjonalnego podejścia do zagadnień technicznych, skrupulatnego zapoznania się z dokumentacją oraz warunkami środowiskowymi, w których urządzenia i ich elementy mogą funkcjonować; - [K_K5]		

### 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 zaliczeniowym o charakterze teoretycznym i praktycznym,
- ii. ocenę wiedzy i umiejętności na podstawie indywidualnego omówienia wyników kolokwium (dodatkowe pytania kontrolne),

b) w zakresie ćwiczeń laboratoryjnych weryfikowanie założonych efektów kształcenia realizowane jest przez:

- i. ocenę przygotowania studenta do poszczególnych ćwiczeń laboratoryjnych oraz ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych (krótki sprawdzian),
- ii. ocenianie ciągle, na każdych zajęciach (odpowiedzi ustne) ? premiowanie przyrostu umiejętności posługiwania się poznanymi metodami i narzędziami.

### Treści programowe

Program przedmiotu:

1. Wprowadzenie do systemów czasu rzeczywistego: definicje ograniczeń czasowych, systemy operacyjne czasu rzeczywistego, działania procesów aplikacyjnych, standard POSIX.
  2. Architektura systemu czasu rzeczywistego: struktura systemu operacyjnego, jądro systemu i jego funkcje, komunikaty i komunikacja międzyprocesowa, administratorzy zasobu i procesy systemowe, system plików, sieć komunikacyjna.
  3. Procesy i wątki: definicje procesu, wątku, szeregowanie wątków, procesy i wątki asynchroniczne, synchroniczne, drugoplanowe, rodzaje szeregowania wątków, model stanów procesu, stany procesów i wątków systemie RT, zarządzanie procesami: atrybuty, tworzenie, obsługa zakończenia, zarządzanie wątkami: procesy wielowątkowe, tworzenie wątków, kończenie, łączenie, anulowanie, atrybuty wątku, priorytet wątku, strategie szeregowania, wyścigi wzajemne wykluczanie, muteksy, inwersja priorytetów, synchronizacja wątków i zmienne warunkowe.
  4. Komunikacja pomiędzy procesami: Pliki, łącza nienazwane, łącza nazwane, Komunikaty: tworzenie kanałów i połączeń, funkcje wysyłania, odbioru i potwierdzania komunikatów, Pamięć dzielona i semafor: funkcje operujące na pamięci dzielonej, semafor nienazwane, nazwane, Sygnały: rodzaje, wysyłanie, maskowanie, obsługa, oczekiwanie na sygnały i ich testowanie.
  5. Pomiar czasu w systemie: układy do pomiaru czasu, czas systemowy, opóźnienia.
  6. Timery i zdarzenia: Tworzenie usuwanie timerów, Obsługa zdarzeń, Przykłady zastosowań.
  7. Kolejki komunikatów: Własności, tworzenie, kasowanie, atrybuty kolejek komunikatów, zapis, powiadomienie i odczyt komunikatu.
  8. Przerwania: Obsługa systemu przerwań, Odpytania, blokowanie, maskowanie przerwań, Procedury obsługi przerwania, przerwania a zdarzenia, Własności czasowe systemu przerwań.
  9. Obsługa transmisji szeregowej.
  10. Przykłady zastosowań systemów czasu rzeczywistego do sterowania i pomiaru.
- Ćwiczenia laboratoryjne są realizowane w zespołach 2 osobowych, które wykorzystują 8 stanowisk wyposażonych w skonfigurowany sprzęt komputerowy z peryferiami, wymagany system operacyjny, narzędzia programistyczne. Zadania laboratoryjne polegają na napisaniu oprogramowania z wykorzystaniem wybranego narzędzia, uruchomieniu i przetestowaniu aż do osiągnięcia poprawności działania. Program zajęć laboratoryjnych obejmuje:
1. Wprowadzenie do środowiska projektowania oprogramowania Visual Studio 2010.
  2. Aplikacja okienkowa wykorzystująca komponenty .NET.
  3. Podstawy aplikacji wielowątkowych w Visual Studio 2010.
  4. Obsługa sekcji krytycznej.
  5. Wprowadzenie do środowiska i programowania systemu QNX.
  6. Synchronizacja procesów.
  7. Synchronizacja wątków.
  8. Przesyłanie komunikatów pomiędzy procesami.
  9. Wykorzystanie timera do generowania zdarzeń.
  10. Wykorzystanie przerwania do generowania zdarzeń.

<b>Literatura podstawowa:</b>		
1. 1.	J. Ulasiewicz, Systemy Czasu rzeczywistego QNX 6.0 Neutrino, Wydawnictwo BTC, Warszawa 2007.	
2. 2.	Programowanie równoległe i asynchroniczne wC# 5.0, Wydawnictwo Helion, Gliwice 2014.	
3. 3.	P. Majdzik, Programowanie współbieżne, Systemy czasu rzeczywistego, Wydawnictwo Helion, Gliwice 2012.	
4. 4.	K. Lal, T. Rak, K. Orkisz, RTLinux - system czasu rzeczywistego, Helion, Gliwice 2003.	
<b>Literatura uzupełniająca:</b>		
1. 1.	A. Silberschatz, P. B. Galvin, G. Gagne, Podstawy systemów operacyjnych, WNT, wyd. 6, Warszawa 2005.	
2. 2.	QNX Momentics IDE User's Guide, 2002?2014, QNX Software Systems Limited.	
3. 3.	QNX Software Development Platform 6.6, 2002?2014, QNX Software Systems Limited.	
<b>Bilans nakładu pracy przeciętnego studenta</b>		
<b>Czynność</b>		<b>Czas (godz.)</b>
1. udział w wykładach		15
2. udział w ćwiczeniach laboratoryjnych:		30
3. przygotowanie do ćwiczeń laboratoryjnych:		10
4. dokończenie (w ramach pracy własnej) zadań z ćwiczeń:		10
5. udział w konsultacjach związanych z realizacją procesu kształcenia: z ćwiczeń audytoryjnych oraz wykładów		2
6. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron		20
7. przygotowanie do pisemnego kolokwium zaliczeniowego z wykładu:		10
8. udział w kolokwium zaliczeniowym		2
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	99	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	49	2
Zajęcia o charakterze praktycznym	50	2