



POLSKO-JAPOŃSKA AKADEMIA TECHNIK KOMPUTEROWYCH

SYLABUS PRZEDMIOTU

Nazwa przedmiotu:	Systemy czasu rzeczywistego
Kod przedmiotu:	SCR
Kierunek / Profil:	Informatyka / praktyczny
Tryb studiów:	stacjonarny
Rok / Semestr:	3 / 5
Charakter:	obieralny
Odpowiedzialny:	dr inż. Władysław Szczepniak,
Wersja z dnia:	19.02.2026

1. Godziny zajęć i punkty ECTS

Wykłady	Ćwiczenia	Laboratoria	Z prowadzącym	Praca własna	Łącznie	ECTS
30 h	30 h	—	60 h	40 h	100 h	4

2. Forma zajęć

Forma zajęć	Sposób zaliczenia
Laboratorium	Zaliczenie z oceną
Wykład	Nieoceniany

3. Cel dydaktyczny

Celem przedmiotu jest zapoznanie studentów z podstawowymi zagadnieniami dotyczącymi systemów czasu rzeczywistego (SCR), dotyczącymi: zasad funkcjonowania i specyfikacji systemów czasu rzeczywistego i systemów operacyjnych czasu rzeczywistego, a także metodologii tworzenia aplikacji czasurzeczywistego. Pozwala to na osiągnięcie praktycznej umiejętności tworzenia aplikacji czasu rzeczywistego dla określonej specyfikacji i wybranego hardware'u.

4. Przedmioty wprowadzające

Przedmiot

Analiza matematyczna, Algebra liniowa i geometria, Fizyka, Użytkowanie komputerów, Wstęp do inf

Znajomość regulaminu i zasad BHP obowiązujących w laboratorium.

5. Treści programowe

1. 1 , Wprowadzenie. Podstawowe definicje. Wymagania stawiane systemom czasu rzeczywistego (SCR). Podział SCR. Wybór algorytmów stosowanych dla potrzeb SCR, wy oceny ich złożoności obliczeniowej.
2. 2 , Dobór algorytmu dla potrzeb SCR. Weryfikacja poprawności logicznej wybranych algorytmów. Symulacyjne szacowanie czasu realizacji zadań algorytmów, które uzyskały pozytywną weryfikację poprawności logicznej. Metody przyspieszania działania SCR. Symulacyjna weryfikacja poprawności logicznej wybranych algorytmów dla potrzeb SCR.
3. 3 , Charakterystyczne cechy, potrzeby i rozwój systemów czasurzeczywistego. Metodaiteracyjnego osiągnięcia wymagań, narzuconych w specyfikacji zleceniodawcy systemu czasu rzeczywistego. Szacowanie czasu realizacji zadań dla algorytmów, które uzyskały pozytywną weryfikację poprawności logicznej, wybranych do realizacji SCR.
4. 4 , Tworzenie zadań i metody ich szeregowania. Wybrane algorytmy szeregowania zadań. Zastosowanie szeregowania zadań do alokacji operacji algorytmów SCR, implementowanych na określonym/dedykowanym hardware. Szeregowanie wieloprocesorowe. Obsługa przerw i sygnałów. Obsługa czasu. Tolerancje czasowe. Synchronizacja procesów. Stosowane protokoły komunikacyjne. Szeregowanie wieloprocesorowe. Szeregowanie w sieciach rozproszonych. Analiza utworzonych algorytmów i ich kwalifikacja do implementacji w systemach czasu rzeczywistego. Metody polepszania parametrów projektowanego SCR (algorytmiczne i programistyczne).
5. 5 , Systemy operacyjny spełniający wymogi czasu rzeczywistego (RTOS). Definicja systemu operacyjnego czasu rzeczywistego i jego podstawowe cechy. Budowa sys-

temu operacyjnego czasu rzeczywistego. Mechanizmy systemów czasu rzeczywistego: algorytmy wyłączenia, semaforey, odmierzenie czasu. Czasy reakcji systemu. Komunikacja międzyzadaniowa w systemie operacyjnym. Powtarzalność czasów reakcji systemu dla tych samych zadań. Analiza przypadków zastosowania systemów operacyjnych czasu rzeczywistego. System operacyjny czasu rzeczywistego QNX RTOS Wybór systemu operacyjnego (RTOS). dla tworzonego SCR.

6. 6 , Przykłady systemów czasu rzeczywistego i wybrane metody ich projektowania dla zadanej specyfikacji. Przypadki nieokreśloności czasu opóźnienia SCR i metody ich eliminacji/redukcji. Podstawy obsługi QNX RTOS.
7. 7 , Przykłady praktycznie działających urządzeń/instalacji technologicznych, spełniających wymogi SCR. Heterogeniczny system czasu rzeczywistego dla potrzeb przemysłowych (monitoring, sterowanie i diagnostyka w rozproszonej sieci rozległej).
8. 8 , Wymagania dotyczące hardware'u dla potrzeb systemów czasu rzeczywistego. Metody porównywania procesorów i ich wybór dla potrzeb SCR. Wybór hardware'u dla projektowanego SCR. Analiza wpływu elementów hardware'u na opóźnienia czasowe. Właściwy dobór częstotliwości próbkowania przetworników A/D - wpływ na czas opóźnienia systemu. Wprowadzenie do wybranego środowiska programistycznego (ang. IDE) dla potrzeb określonych procesorów. Obsługi wybranego środowiska i zasady programowania. Program zaimplementowany na wybranym hardware.
9. 9 , Języki programowania stosowane do tworzenia SCR. Programowania urządzeń wbudowanych SCR. Automatyczna generacja kodu (kiedy występują takie możliwości). Realizacja wybranego projektu SCR, np. sterowanie światłami w ruchu ulicznym.
10. 10 , Potrzeba redukcji poboru mocy w SCR Ocena jakości utworzonego kodu programu dla SCR.
11. 11 , Metody optymalizacji kodu dla potrzeb SCR. Walidacja utworzonego SCR. Analiza potrzeb dodatkowych optymalizacji. Wielokrotna optymalizacja SCR. ,
12. 12 , Szybkie tworzenie prototypu systemu czasu rzeczywistego, np. z wykorzystaniem układów FPGA. Projektowanie i symulacje zadanych SCR. Pomiar czasu opóźnienia systemu.
13. 13 , Zagadnienia bezpieczeństwa w sieciach rozproszonych SCR.
14. 14 , Analiza utworzonego systemu z punktu widzenia spełnienia wymagań dotyczących jego specyfikacji.
15. 15 , Zastosowania wybranych rozwiązań SCR w różnych dziedzinach, np. automatyka przemysłowa, robotyka, internet rzeczy, logistyka dedykowanych dostaw towarów, medycyna, technologie kosmiczne, technologie militarne i inne.

6. Efekty kształcenia

Wiedza

- Student ma uporządkowaną wiedzę w zakresie architektur komputerów, systemów i sieci komputerowych oraz systemów operacyjnych w tym systemów operacyjnych czasu rzeczywistego.
- Student ma podstawową wiedzę w zakresie architektur i programowania systemów

mikroprocesorowych, wybranych języków mikroprogramowania procesorów, zna i rozumie zasadę działania podstawowych modułów peryferyjnych oraz interfejsów komunikacyjnych stosowanych w systemach mikroprocesorowych.

- Student ma uporządkowaną, podbudowaną teoretycznie wiedzę w zakresie systemów operacyjnych i zasad ich działania, współbieżności i szeregowania zadań, metod synchronizacji i komunikacji między procesami.

Umiejętności

- Student potrafi skonstruować algorytm rozwiązania prostego zadania inżynierskiego oraz zaimplementować, przetestować i uruchomić go w wybranym środowisku programistycznym na komputerze klasy PC dla wybranych systemów operacyjnych.
- Student potrafi dobrać rodzaj i parametry układu wykonawczego, układu pomiarowego, jednostki sterującej oraz modułów peryferyjnych i komunikacyjnych dla wybranego zastosowania oraz dokonać ich integracji w postaci wynikowego systemu pomiarowo-sterującego.
- Student potrafi zaplanować, przygotować i przeprowadzić symulację działania prostych układów automatyki i robotyki.
- Student potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny, używając właściwych metod, technik i narzędzi.

Kompetencje społeczne

- Student posiada świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżynierskiej w tym jej wpływ na środowisko i związaną z tym odpowiedzialność za podejmowane decyzje.

7. Kryteria oceny

- wykład z elementami dyskusji z prezentacją multimedialną
- projektowanie systemów dla zadanej specyfikacji
- rozwiązywanie zadań projektowych
- analiza przypadków
- badania symulacyjne
- Kryteria oceny
- Sprawdziany, tzw. wejściówki (średnia za semestr - awe)
- ocena z aktywności na zajęciach (średnia za semestr - apr)
- ocena końcowego zaliczenia sprawozdań- kz_sp)
- Końcowa ocena zaliczenia laboratorium (zl) jest wyznaczana na podstawie poniższego wzoru:
- $zl = (2.0 * awe + 1.9 * apr + 1.1 * kz_sp) / 5$
- Wykład (ocena z wykładu zw) jest zaliczany na podstawie dwóch kartków (k1, k2)
- Każda nieobecność na wykładzie, rozpoczynając od drugiej powoduje obniżenie oceny końcowej zw o wartość 0,1.

- Ocena końcowa (zal) zaliczająca przedmiot (SCR) jest wyliczana wg poniższego wzoru:
- przy czym obie oceny składowe muszą być pozytywne.

8. Metody dydaktyczne

Wykład, laboratoria, praca własna studenta.

9. Literatura

Podstawowa:

- M. Chetto. Ed. Real-time Systems Scheduling 1. Fundamentals. J. Wiley & Sons, 2014.
- A. Gupta, A. Kumar Chandra and P. Luksch. Real-Time and Distributed Real-Time Systems. Theory and Applications. CRC Press 2016.
- H. Kopetz. Real - Time Systems. Springer 2020.
- A. Kwiecień, P. Gaj. (Red.). Współczesne problemy systemów czasu rzeczywistego. WNT, Warszawa 2004.
- P. Laplante and S. Ovaska. REAL-TIME SYSTEMS DESIGN AND ANALYSIS. Tools for the Practitioner. Fourth Edition. IEEE Press & J. Wiley 2012.
- C. Ptolemaeus, Ed. System Design, Modeling, and Simulation using Ptolemy II. Ptolemy org, 2014.
- K. Sacha. Systemy czasu rzeczywistego. Wyd. PW 2006.
- J. Ułasiewicz. Systemy czasu rzeczywistego QNX6 Neutrino, Wyd. BTC, Legionowo 2007.

Uzupełniająca:

- W. Kaczmarek. Programowanie robotów przemysłowych. PWN, Warszawa 2017.
- M. Ben-Ari. Podstawy programowania współbieżnego i rozproszonego. WNT, 2009.
- Elektronika Praktyczna. Miesięcznik AVT. Wybrane artykuły.
- Automatyka, Podzespoły i Aplikacje. Miesięcznik AVT. Wybrane numery.