**Nazwa przedmiotu:**

Systemy czasu rzeczywistego

**Koordynator przedmiotu:**

Krzysztof SACHA

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia II stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne - podstawowe

**Kod przedmiotu:**

SCZR

**Semestr nominalny:**

1 / rok ak. 2015/2016

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

100
- 30 godzin wykładu
- 15 godzin ćwiczeń laboratoryjnych
- 5 godzin konsultacji
- 50 godzin samodzielnej pracy związanej z przygotowaniem rozwiązania zadań laboratoryjnych oraz przygotowaniem do kolokwium

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

2

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

2

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 0h |
| Laboratorium: | 15h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Elementarna znajomość techniki cyfrowej, programowania w języku C oraz zagadnień systemów operacyjnych

**Limit liczby studentów:**

120

**Cel przedmiotu:**

Celem przedmiotu jest przedstawienie specyfiki i zakresu zastosowania systemów czasu rzeczywistego, wyjaśnienie metodyki projektowania i programowania systemów wbudowanych oraz wprowadzenie do zagadnień bezpieczeństwa i niezawodności systemu. Wykład obejmuje także systemy operacyjne czasu rzeczywistego i sieci przemysłowe. Ćwiczenia laboratoryjne pozwalają studentom nabyć praktyczną umiejętność projektowania, programowania i integracji rozproszonych aplikacji czasu rzeczywistego.

**Treści kształcenia:**

Wprowadzenie (2h). Charakterystyka i zakres zastosowania systemów czasu rzeczywistego, model działania systemu, zadania taktowane czasem i taktowane zdarzeniami, klasyfikacja ograniczeń czasowych. Przykłady. Proces tworzenia oprogramowania systemu wbudowanego.
Sterownik wbudowany (4h). Wymagania stawiane sterownikom wbudowanym. Sprzęg procesowy, podstawowe bloki funkcjonalne, sygnały analogowe i dwustanowe, sterowanie PWM, optoizolacja. Przykłady: karta PCL-718, programowa obsługa układów sprzęgu procesowego. Komputery kasetowe i panelowe. Mikrokontrolery scalone, przykłady: 80C51, ARM7TDMI, metody oszczędzania energii.
Organizacja oprogramowania (4h). Cykliczne odpytywanie (polling), ręczne szeregowanie zadań, szeregowanie kooperatywne, wykorzystanie przerwań, systemy dwuplanowe (foreground/background), problem synchronizacji zadań. Wielozadaniowy system operacyjny. Przykłady: sterownik PLC, kontroler nadrzędny.
Przykład implementacyjny (2h). Instalacja sterowana i wymagania użytkownika, określenie architektury systemu. Analiza wymagań dla oprogramowania, określenie ograniczeń czasowych. Projekt struktury oprogramowania, implementacja cyklicznego programu sterującego.
Szeregowanie zadań (1h). Problem szeregowania zadań, algorytmy RMS i EDF, warunki dostateczne szeregowalności, przewidywalność działania przy przeciążeniu. Algorytm serwera sporadycznego. Wpływ synchronizacji na szeregowanie, inwersja priorytetu.
System operacyjny czasu rzeczywistego (4h). Specyfika zadań czasu rzeczywistego. Model budowy systemu, graf stanów zadania, tablica stanu, algorytmy podstawowych operacji podziału czasu procesora. Synchronizacja i komunikacja zadań, procesy i wątki, zakleszczenia, inwersja priorytetów. Przegląd systemów i platform sprzętowych.
API programisty (4h). Standard POSIX, tworzenie i usuwanie procesów i wątków, algorytmy szeregowania, sygnały i sygnały kolejkowane, zdarzenia, zarządzanie pamięcią. Narzędzia synchronizacji i komunikacji zadań, anomalia synchronizacji, protokoły obsługi priorytetu. Uzależnienia czasowe. Asynchroniczne operacje we/wy. Przykład: system QNX.
Sieci przemysłowe (3h). Wymagania stawiane sieciom przemysłowym, protokoły warstwy drugiej (liniowej), przykłady popularnych sieci przemysłowych. Sieć Profibus: warstwa fizyczna, protokoły i usługi warstwy liniowej, usługi warstwy aplikacyjnej (FMS), wykonanie usług w relacjach komunikacyjnych różnego typu, API programisty. Ethernet przemysłowy.
Bezpieczeństwo systemów (2h). Norma IEC 61508, identyfikacja zagrożeń, ocena ryzyka. Analiza bezpieczeństwa metodą FTA, systemy zabezpieczeń. Poziomy nienaruszalności, rekomendacje normy IEC 61508.
Studium przypadku (2h). System automatyzacji budynku, architektura systemu, sieci, sterowniki PLC, programowanie graficzne, stacja operatora procesu. Sterownik regionalny, zadania, architektura oprogramowania.
Kolokwia zaliczeniowe (2h).

**Metody oceny:**

Wykład: kolokwium zaliczeniowe.
Laboratorium: ocena wyników pracy studenta.
Wymagane zaliczenie obydwu części przedmiotu.

**Egzamin:**

nie

**Literatura:**

1. Sacha K., Systemy czasu rzeczywistego, Wyd. 2 (zmienione), Oficyna Wydawnicza PW, Warszawa, 1999.
2. Sacha K., Sieci przemysłowe - Profibus, Mikom, 1998.
3. Phillip A. Laplante, Real-Time Systems Design and Analysis, John Wiley & Sons, 2004

**Witryna www przedmiotu:**

brak

**Uwagi:**

Przedmiot prowadzony w każdym semestrze (letnim i zimowym).
Przeznaczony dla specjalności I-SID

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt SCZR\_W01:**

Ma szczegółową wiedzę w zakresie systemów i sieci przemysłowych oraz zagadnień bezpieczeństwa

Weryfikacja:

Kolokwium

**Powiązane efekty kierunkowe:** K\_W04

**Powiązane efekty obszarowe:** T2A\_W02

**Efekt SCZR\_W02:**

Ma szczegółową wiedzę na temat architektury oprogramowania systemów sterujących i technologii systemów operacyjnych czasu rzeczywistego

Weryfikacja:

Kolokwium

**Powiązane efekty kierunkowe:** K\_W10

**Powiązane efekty obszarowe:** T2A\_W03, T2A\_W04, T2A\_W05, T2A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt SCZR\_U01:**

Potrafi ocenić możliwość zastosowania różnych architektur oprogramowania w zależności od istniejących uwarunkowań czasowych

Weryfikacja:

Kolokwium + zaliczenie laboratorium

**Powiązane efekty kierunkowe:** K\_U13

**Powiązane efekty obszarowe:** T2A\_U18