

KARTA PRZEDMIOTU

Nazwa przedmiotu: **Systemy wbudowane i systemy czasu rzeczywistego (AiRAu-R>SM1-SW-19)**

Nazwa w języku polskim:

Nazwa w jęz. angielskim: **Embedded and real-time systems**

Dane dotyczące przedmiotu:

Jednostka oferująca przedmiot: Wydział Automatyki, Elektroniki i Informatyki

Przedmiot dla jednostki: Politechnika Śląska

Domyślny typ protokołu dla przedmiotu:

ZAL

Język wykładowy:

polski

Strona WWW:

<https://platforma2.polsl.pl/rau1/course/view.php?id=277>

Skrócony opis:

Celem przedmiotu jest zapoznanie słuchaczy z zagadnieniami związanymi z budową układów mikroprocesorowych ze szczególnym uwzględnieniem wykorzystania ich w systemach wbudowanych oraz wykorzystania możliwości systemów czasu rzeczywistego do zarządzania procesami. Wiedza nabyta w trakcie wykładów pozwala na projektowanie elektronicznych układów mikroprocesorowych jako układów kontrolno-pomiarowych (telemetria, interfejsy użytkownika) oraz układów sterowania wykorzystujących różnego typu czujniki działające w oparciu o systemy czasu rzeczywistego

Opis:

Wykład:

Wstęp i wprowadzenie do tematyki systemów wbudowanych i systemów czasu rzeczywistego. Omówienie typów mikrokontrolerów z naciskiem na podział SoC (ang. System on a Chip) i SoM (ang. System on a Module). Wymagania stawiane systemom mikroprocesorowym – koszty, wielofunkcyjność, elastyczność.

Oprogramowanie narzędziowe: oprogramowanie uruchomieniowe, emulowanie procesora w układzie – ICE (In Circuit Emulator), śledzenie wykonywania programu – ICD (In Circuit Debugger), programowanie procesorów – programowanie w układzie (In Circuit Serial Programming), wykorzystanie oprogramowania ładującego (Bootloader), uruchamianie aplikacji – podstawowe procedury uruchomieniowe, konfigurowanie procesora.

Układy wejścia i wyjścia w systemach wbudowanych: magistrale I2C, SPI, CAN, 1-wire, USART, przetworniki A/C i C/A, wpływ typu przetwornika A/C na dokładność pomiaru sygnału wejściowego.

Pojęcia związane z systemami operacyjnymi czasu rzeczywistego (RTOS). Wymagania stawiane RTOS. Systemy operacyjne czasu rzeczywistego i ich krótka charakterystyka (Linux z PREEMPT_RT, FreeRTOS).

RTOS na tle typowych systemów operacyjnych ogólnego użytku. System operacyjny Linux z PREEMPT_RT jako przykład RTOS. Budowa wewnętrzna systemu FreeRTOS. Systemy operacyjne z jądrem monolitycznym oraz oparte na mikrojądrze.

Zadania jądra systemu. Proces i jego atrybuty. Algorytmy szeregowania procesów. Mechanizmy komunikacji międzyprocesowej. Mechanizmy synchronizacji dostępu do pamięci dzielonej. Blokady. Niekontrolowana inwersja priorytetów.

Tworzenie sterowników urządzeń oraz aplikacji dla systemów czasu rzeczywistego. Nie determinizm czasowy związany z niektórymi cechami architektury mikroprocesora oraz systemu (wpływ Cache, MMU, mechanizmów typu SMI niewidzialnych dla jądra systemu operacyjnego).

Systemy RTOS w automatyce i robotyce.

Laboratorium:

1. Uruchomienie systemu wbudowanego typu SoC i SoM.
2. Uruchomienie, śledzenie i usuwanie błędów w programie z wykorzystaniem modułu ICD na przykładzie modułów SoC i SoM.
3. Obsługa zewnętrznych czujników w systemach wbudowanych.
4. Komunikacja w systemach wbudowanych.
5. Instalacja systemu czasu rzeczywistego FreeRTOS i RT Linux na przykładowych systemach wbudowanych.
6. Implementacja algorytmów sterowania w oparciu o system RTOS.
7. Ocena przydatności systemów RTOS w automatyce i robotyce.

Liczba godzin zajęć z bezpośrednim udziałem nauczycieli akademickich:

Liczba punktów ECTS: 4

Całkowita liczba godzin: 100 (kontaktowo 60h / praca własna 40h)

Wykład: 30h

Ćwiczenia laboratoryjne: 30h

Praca własna studenta: przygotowanie do zaliczenia wykładu (20h), przygotowanie do realizacji ćwiczenia laboratoryjnego (20h)

w tym

Liczba punktów ECTS uzyskanych na zajęciach z bezpośrednim udziałem nauczyciela akademickiego: 1

Liczba punktów ECTS uzyskanych na zajęciach o charakterze praktycznym (laboratoria, projekty): 3

Literatura:

1. Pietraszek St., Mikroprocesorowe jednoukładowe PIC, Helion 2002.

2. Iovine J., PIC Robotics, McGraw-Hill 2004.
3. Kurfess T., Robotics and Automation Handbook, CRC Press 2005.
4. Rzeszut P., Systemy embedded w FPGA, Wydawnictwo BTC 2019.
5. Silberschatz A., Galvin P.B., Gagne G., „Podstawy systemów operacyjnych”, wyd. 6 zmienione i rozsz., WNT 2005.
6. Predko M., Programming and Customizing PICmicro Microcontrollers, MIT Press 2000.
7. Rubini A., Corbet J.: „Linux Device Drivers. 2nd Edition”. O'Reilly & Associates. Inc.. 2001.

Efekty uczenia się:

K2A_W07 - zasady działania i metody konfiguracji systemów operacyjnych czasu rzeczywistego, systemów rozproszonych oraz przemysłowych sieci i baz danych

K2A_W08 - zasady projektowania, konfigurowania i programowania platform sprzętowych, w tym sterowników przemysłowych i dedykowanych, dla celów przetwarzania informacji i sterowania

K2A_U08 - biegle posługiwać się właściwie dobranymi środowiskami i narzędziami programistycznymi do symulacji, projektowania i oceny jakości złożonych systemów automatyki i robotyki, określić ich istotne parametry i charakterystyki eksploatacyjne

K2A_U19 - przy formułowaniu, rozwiązywaniu i realizacji zadań, związanych z projektowaniem i modelowaniem układów i systemów automatyki, robotyki i mechatroniki oraz oceną przydatności i możliwości wykorzystania nowych osiągnięć (technik i technologii) integrować wiedzę z automatyki i robotyki, informatyki, mechaniki, biotechnologii, sztucznej inteligencji oraz innych dyscyplin i dziedzin wiedzy

Metody i kryteria oceniania:

Wykład

Kolokwium pisemne składające się z pięciu pytań opisowych.

Kryterium zaliczenia: minimum 60% poprawnych odpowiedzi.

Ćwiczenia laboratoryjne

Zaliczenie pisemne w postaci raportu z wykonanego ćwiczenia laboratoryjnego.

Kryterium zaliczenia: dostarczenie raportu z wynikami sprecyzowanego zadania laboratoryjnego.

Sylabus obowiązuje od roku akademickiego 2024/2025, a jego zawartość nie podlega zmianom w trakcie trwania semestru.

Punkty przedmiotu w cyklach:

<bez przypisanego programu>			
Typ punktów	Liczba	Cykl pocz.	Cykl kon.
Europejski System Transferu Punktów (ECTS)	4	2020/2021-L	