

KARTA PRZEDMIOTU OFEROWANEGO W SZKOLE DOKTORSKIEJ

Kod przedmiotu	4606-PS-000000B-0252	Nazwa przedmiotu	w j. polskim	Systemy wbudowane i sterowniki (SWIS)	
			w j. angielskim	Embedded systems and device drivers	
Przynależność do grupy przedmiotów	przedmioty specjalnościowe				
Koordinator przedmiotu	dr hab. inż. Wojciech Zabołotny				
Jednostka realizująca	WEiTI	Dyscyplina/y naukowa*	Automatyka Elektronika, Elektrotechnika i Technologie Kosmiczne		
Poziom kształcenia	Kształcenie doktorantów	Semestr	zimowy/letni		
Język zajęć	polski				
Forma zaliczenia:	egzamin	Sumaryczna liczba godzin w semestrze	60	Sumaryczna liczba ECTS	4
Minimalna liczba uczestników	10	Maksymalna liczba uczestników	30	Dostępność dla studentów	Tak/Nie
Typ zajęć		Wykład	Ćwiczenia audytoryjne	Ćwiczenia projektowe	Laboratorium
Liczba godzin zajęć	tygodniowo	2	-	1	1
	łącznie w semestrze	30	-	15	15

* nie dotyczy warsztatu badacza

1. Wymagania wstępne

Wymagane jest, aby słuchacz posiadał umiejętność programowania w języku C. Słuchacz powinien także znać zagadnienia związane z zasadami działania i architekturą mikroprocesorów i systemów komputerowych. Wymagana jest znajomość technicznych aspektów systemów operacyjnych oraz tworzenia niskopoziomowych aplikacji ściśle współpracujących z systemem operacyjnym. Zalecana jest znajomość zagadnień związanych programowalnymi układami logicznymi, w tym umiejętność realizacji systemów cyfrowych w układach FPGA z wykorzystaniem języków HDL. Korzystanie z wykorzystywanych na przedmiocie źródeł literaturowych wymaga znajomości języka angielskiego.

2. Cele przedmiotu

We współczesnych systemach wbudowanych konieczność obsługi specjalizowanego sprzętu łączy się z wykorzystaniem standardowych systemów operacyjnych, pozwalających na realizację złożonego oprogramowania sterującego i przetwarzającego dane. Dlatego projektowanie i realizacja takich systemów wymaga zrozumienia działania interfejsów używanych do komunikacji z urządzeniami peryferyjnymi oraz obsługujących je mechanizmów dostarczanych przez system operacyjny. Ukończenie kursu powinno zapewnić studentom teoretyczną wiedzę i praktyczne umiejętności, pozwalające tworzyć, uruchamiać i testować sterowniki urządzeń działające w przestrzeni jądra oraz współpracujące z nimi aplikacje. Przedmiot uwzględnia zagadnienia związane z bezpieczeństwem i wydajnością stosowanych rozwiązań. Praktyczne aspekty przedmiotu realizowane są na zintegrowanych zajęciach projektowo-laboratoryjnych, na których studenci będą mogli najpierw oswoić się z poznawaną dziedziną, analizując i testując gotowe rozwiązania, następnie rozpocząć samodzielne ich modyfikowanie, a w końcu zaprojektować i zrealizować projekt (w miarę możliwości zespołowy) obejmujący stworzenie urządzenia wraz z jego sterownikiem i obsługującą je aplikacją. Urządzenie może zostać zrealizowane jako model symulacyjny, jako system mikrokontrolerowy, lub jako blok IP realizowany w logice programowalnej systemu SoC (z wykorzystaniem języków HDL i/lub syntezy wysokopoziomowej HLS). Podstawowym systemem operacyjnym omawianym na wykładach i wykorzystywanym na zajęciach projektowo-laboratoryjnych jest Linux. Istnieje jednak możliwość realizacji projektu związanego z innym systemem (np. Windows, Android, Zephyr itp.)

3. Treści programowe (dla każdego typu zajęć oddzielnie)

Wykład

1. Mechanizmy współczesnych systemów operacyjnych wspomagające wydajną i bezpieczną komunikację z urządzeniami peryferyjnymi.
2. Interfejsy sprzętowe używane do komunikacji z urządzeniami peryferyjnymi.
 1. Proste interfejsy - I2C, SPI, UART
 2. Interfejsy o złożonej obsłudze (np. Bluetooth, USB).
 3. Interfejsy o dużej wydajności - AXI, PCIe itp.
3. Metody tworzenia sterowników urządzeń
 1. Metody komunikacji sterownika z aplikacjami
 2. Funkcje związane z obsługą interfejsów komunikacyjnych (w tym obsługa przerwań)
 3. Zarządzanie pamięcią i komunikacja z pamięcią (DMA)
 4. Aspekty związane z pracą w czasie rzeczywistym
4. Realizacja urządzeń peryferyjnych współpracujących ze sterownikami
 1. Układy specjalizowane
 2. Realizacja z wykorzystaniem mikrokontrolerów
 3. Realizacja w postaci modelu symulacyjnego na potrzeby testów
 4. Realizacja w logice programowalnej z wykorzystaniem języków opisu sprzętu (Verilog/VHDL) a także syntezy wysokopoziomowej (HLS).
 1. Techniki uruchamiania i testowania urządzeń i sterowników w systemach SoC
5. Realizacja aplikacji współpracujących ze sprzętem
 1. Techniki właściwego wykorzystania interfejsu sterownika
 2. Możliwości bezpośredniej współpracy ze sprzętem - zalety, wady i ograniczenia
6. Realizacja kompletnego systemu
 1. Podział funkcji między sprzęt, sterowniki działające w przestrzeni jądra i aplikacje działające w przestrzeni użytkownika
 2. Wykorzystanie możliwości przenoszenia funkcji między sprzętem, sterownikami i aplikacjami w celu zapewnienia wydajnego uruchamiania.
 3. Zagadnienia związane z niezawodnością i energooszczędnością systemu

Laboratorium

W ramach zajęć projektowo-laboratoryjnych słuchacze będą mieli możliwość realizacji następujących zadań:

1. Poznanie i testowanie gotowych sterowników urządzeń wirtualnych
2. Poznanie i testowanie sterowników urządzeń zrealizowanych jako model w symulatorze (np. w QEMU).
3. Poznanie i testowanie sterowników urządzeń podłączonych przez interfejsy SPI, I2C itp.
4. Poznanie i testowanie sterowników urządzeń zrealizowanych w logice programowalnej w układzie SoC i podłączonych do magistrali procesora.

Ponadto w ramach zadań projektowo-laboratoryjnych studenci będą zobowiązani do zrealizowania kompletnego systemu (zalecana jest praca zespołowa), wykorzystującego samodzielnie zaproponowane urządzenie peryferyjne (w postaci modelu dla symulatora, w postaci urządzenia

mikrokontrolerowego, lub w postaci systemu realizowanego w logice programowalnej)), opracowany do niego sterownik urządzenia i współpracujące z nim aplikacje użytkownika.

4. Efekty uczenia się			
Rodzaj efektu	Opis efektu uczenia się	Odniesienie do efektów uczenia się w SD PW	Sposób weryfikacji efektów uczenia*
Wiedza			
W01	Słuchacz ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną obejmującą kluczowe zagadnienia w zakresie systemów mikroprocesorowych i systemów wbudowanych	SD_W2	Egzamin, ocena projektu, ocena raportu
W02	Słuchacz zna i rozumie problemy występujące przy realizacji systemów wbudowanych wraz z obsługującym je oprogramowaniem systemowym. Zna występujące ograniczenia i zdaje sobie sprawę z konieczności godzenia niejednokrotnie sprzecznych wymagań wobec takich systemów.	SD_W2	Egzamin, ocena projektu, ocena raportu
W03	Słuchacz rozumie potrzebę śledzenia najnowszych rozwiązań w dziedzinie systemów wbudowanych i ich oprogramowania systemowego i zna aktualne kierunki rozwoju tej dziedziny i dostępne rozwiązania.	SD_W3	Egzamin, ocena projektu, ocena raportu
Umiejętności			
U01	Słuchacz potrafi wykorzystywać wiedzę z różnych dziedzin do twórczego identyfikowania, formułowania i innowacyjnego rozwiązywania złożonych problemów, potrafi rozwijać metody, techniki i narzędzia badawcze oraz twórczo je stosować.	SD_U1	Ocena projektu, ocena raportu
U02	Słuchacz potrafi zebrać niezbędne informacje z dostępnej literatury anglojęzycznej i wykorzystać je do rozwiązania postawionego przed nim problemu.	SD_U6	Egzamin, ocena projektu, ocena raportu
U03	Słuchacz umie zaplanować i zrealizować indywidualnie lub w małym zespole twórcze zadanie polegające na realizacji systemu wbudowanego wraz ze współpracującym sterownikiem i aplikacją, realizujące założone wymagania	SD_U7	Ocena projektu, ocena raportu
Kompetencje społeczne			
K01	Słuchacz potrafi myśleć i działać w sposób kreatywny i przedsiębiorczy.	SD_K4	Ocena projektu

* dozwolone sposoby weryfikacji efektów uczenia się: egzamin; egzamin ustny; kolokwium pisemne; kolokwium ustne; ocena projektu; ocena sprawozdania; ocena raportu; ocena prezentacji; ocena aktywności w trakcie zajęć; prace domowe; test

5. Kryteria oceny

Przedmiot będzie zaliczany na podstawie zajęć laboratoryjno-projektowych i egzaminu końcowego. W ramach części laboratoryjnej rozwiązywane jest 5 zagadnień po 6 punktów (maksymalnie 30 punktów). Realizacja projektu (najlepiej w zespołach 2-osobowych) może dać maksymalnie 40 punktów.

Egzamin końcowy oceniany jest w zakresie od 0 do 30 punktów.

Ocena końcowa wystawiana jest na podstawie sumy zdobytych punktów, przy czym konieczne jest zdobycie przynajmniej 50 punktów.

Skala ocen: 90-100 pkt: 5.0, 80-89.9 pkt: 4.5, 70-79.9 pkt: 4.0, 60-69.9 pkt: 3.5, 50-59.9 pkt: 3.0, poniżej 50 pkt: 2.0

6. Literatura

Literatura podstawowa:

[1] John Madieu, Linux Device Driver Development - Second Edition, Packt Publishing, 2022 (ewentualnie pierwsze wydanie - Madieu, John. *Linux Device Drivers Development*, Packt Publishing, 2017. dostępne przez BG PW w ProQuest Ebook Central)

[2] Abbott, Doug. *Linux for Embedded and Real-Time Applications*, Elsevier Science & Technology, 2012. (dostępne przez BG PW w ProQuest Ebook Central)

[3] Giometti, Rodolfo. *GNU/Linux Rapid Embedded Programming*, Packt Publishing, Limited, 2017. (dostępne przez BG PW w ProQuest Ebook Central)

Literatura uzupełniająca:

[1] Gay, Warren, Beginning STM32: Developing with FreeRTOS, Libopenm3 and GCC, Berkeley, CA: Apress L. P, 2018 (dostępne przez BG PW w Springer Link)

[2] J. Kong, FreeBSD Device Drivers: A Guide for the Intrepid, No Starch Press, Incorporated 2012

[3] D. Koch, F. Hannig, FPGAs for Software Programmers, Springer International Publishing AG 2016

7. Nakład pracy doktoranta niezbędny do osiągnięcia efektów uczenia się**

Lp.	Opis	Liczba godzin
1	godziny kontaktowe z nauczycielem akademickim wynikające z planu	60
2	Godziny kontaktowe z nauczycielem akademickim w ramach konsultacji, egzaminów, sprawdzianów itp.	5
3	Godziny pracy samodzielnej doktoranta w ramach przygotowania do zajęć oraz opracowania sprawozdań, projektów, prezentacji, raportów, prac domowych	35
4	godziny pracy samodzielnej doktoranta w ramach przygotowania do egzaminu, sprawdzianu, zaliczenia	10
Sumaryczny nakład pracy doktoranta		110
Liczba punktów ECTS		4

** 1 ECTS pracy = 25-30 godzin nakładu pracy doktoranta (np. 2 ECTS = 60 godzin; 4 ECTS = 110 godzin)