



**POLITECHNIKA
BYDGOSKA**

Wydział Telekomunikacji,
Informatyki i Elektrotechniki

Karta przedmiotu
Projektowanie i programowanie systemów wbudowanych

1. Informacje podstawowe

Kierunek studiów telekomunikacja i technologie internetu rzeczy		Cykl kształcenia (nabór) 2024/25	
Jednostka zarządzająca kierunkiem studiów Wydział Telekomunikacji, Informatyki i Elektrotechniki		Kod przedmiotu 05TTIRS.PI4C.1382.24	
Poziom studiów pierwszego stopnia (inż.)		Języki wykładowe polski	
Profil studiów Profil ogólnoakademicki		Obligatoryjność Obowiązkowy	
Forma studiów studia stacjonarne		Blok zajęciowy Przedmioty kierunkowe	
Wymagania wstępne		Budowa układów mikroprocesorowych, charakterystyka mediów transmisyjnych, podstawy programowania w językach wysokiego i niskiego poziomu, podstawy elektroniki analogowej i cyfrowej.	
Przedmioty wprowadzające		Podstawy telekomunikacji, Podstawy elektroniki, Programowanie, Protokoły komunikacyjne	
Koordynator		Łukasz Saganowski	
Okres Semestr 3		Forma zaliczenia Zaliczenie na ocenę	Liczba punktów ECTS 4.0
		Forma prowadzenia i godziny zajęć Wykład: 30 Ćwiczenia laboratoryjne: 30	

2. Efekty uczenia się dla przedmiotu

Kod	Opis efektów uczenia się	Odniesienie do kierunkowych efektów uczenia się	Odniesienie do charakterystyk PRK
Wiedza:			
W1	Ma wiedzę na temat narzędzi i środowisk wytwarzania oprogramowania.	TTIR_O1_K_W07	P6S_WG P6S_WG_inż
W2	Zna i rozumie zaawansowane teorie stanowiące podstawę działania technologii cyfrowych, sprzętu, sieci komputerowych i urządzeń z nimi współpracujących.	TTIR_O1_K_W11	P6S_WG P6S_WG_inż
Umiejętności:			
U1	Potrafi ocenić, czy konkretny produkt, usługa lub technologia jest dopasowana do wymaganej specyfikacji	TTIR_O1_K_U02	P6S_UW P6S_UW_inż
U2	Potrafi sprostać nieustannym zmianom wewnętrznym sektora IT i jego otoczeniu zewnętrznym	TTIR_O1_K_U08	P6S_UU
Kompetencje społeczne:			
K1	W sposób krytyczny dokonuje analizy proponowanych rozwiązań, jednocześnie wskazując na konieczność uzupełnienia posiadanej wiedzy w temacie rozwiązywanego problemu	TTIR_O1_K_K01	P6S_KK

3. Treści programowe

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
1.	- Wprowadzenie do systemów wbudowanych definicja, przykłady zastosowań. - Budowa sprzętowa systemów wbudowanych. - Charakterystyka wybranych mikrokontrolerów, mikroprocesorów, układów SoC stosowanych do projektowania systemów wbudowanych np.: ARM, AVR, SoC ARM np.: Sitara, MIPS itp. - Systemy wbudowane oparte o programowalne układy cyfrowe FPGA. - Wprowadzenie do wybudowanych systemów operacyjnych. - Charakterystyka i porównanie wybranych wbudowanych systemów operacyjnych takich jak np.: Embedded Linux, QNX Neutrino, FreeRTOS, NutOS itp. - Budowa wbudowanego systemu operacyjnego na podstawie Debian Linux. Zastosowania wbudowanego systemu operacyjnego Debian Linux • Proces bootowania systemu Linux na podstawie wybranego systemu wbudowanego np: BBB - Beagle Bone Black, • Sposoby podłączania elementów elektronicznych takich jak: rezystor, fotorezystor, tranzystor, fototranzystor, dioda LED, bramki logiczne, czujniki wykorzystujące magistrale I2C, SPI do systemu wbudowanego opartego o Linux np. Beagle Bone Black, • Sposoby obsługi programowej GPIO, przetwornika ADC, watchdoga oraz magistral szeregowych I2C, SPI, UART, PWM, CAN w systemie wbudowanym z wykorzystaniem sterowników wbudowanych w Linux, • Sposoby obsługi urządzeń w systemie wbudowanym Linux, • Kompilacja skrótna (ang. Cross-compilation) aplikacji Linux dla potrzeb systemów wbudowanych opartych o architekturę ARM • Zastosowania systemu wbudowanego Linux w aplikacjach Internetu Rzeczy (ang. Internet of Things), • Omówienie narzędzi programistycznych służących do tworzenia aplikacji dla systemów wbudowanych Linux. • Koprocесory czasu rzeczywistego (PRU-ICSS) Programmable Real-Time Unit and Industrial Communication Subsystem w systemach wbudowanych. - Protokoły teleinformatyczne stosowane do komunikacji pomiędzy systemami wbudowanymi. - Zastosowania sieci telemetrycznych w systemach wbudowanych i przemyśle, transporcie miejskim, transporcie kolejowym, pomiaru wielkości elektrycznych i nieelektrycznych, nadzór nad stanem urządzeń technicznych itp. - Podstawy projektowania sprzętu, obwodów drukowanych, doboru elementów elektronicznych dla systemów wbudowanych.	Wykład	W1, W2

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
2.	• Instalacja systemu wbudowanego Debian Linux na podstawie platformy BBB - Beagle Bone Black, • Konfiguracja zdalnego połączenia VNC (serwer/klient), SSH, wymiana plików, zapoznanie się z strukturą katalogów, uruchamianie programów w systemie wbudowanym Debian Linux, • Kompilacja skrośna oprogramowania z platformy opartej na procesorze x86 na architekturę ARM na przykładzie systemu wbudowanego BBB, • Obsługa portów GPIO za pomocą instrukcji z terminala i skryptu BASH w systemie wbudowanym Debian Linux - BBB, • Obsługa komunikacji szeregowej z wykorzystaniem UART za pomocą instrukcji z terminala i wybranego języka programowania np.: C/C++ w systemie wbudowanym Debian Linux, • Obsługa magistrali szeregowej I2C za pomocą instrukcji z terminala i wybranego języka programowania np.: C/C++ w systemie wbudowanym Debian Linux, • Obsługa magistrali szeregowej SPI za pomocą instrukcji z terminala i wybranego języka programowania np.: C/C++ w systemie wbudowanym Debian Linux, • Obsługa przetwornika ADC za pomocą instrukcji z terminala i wybranego języka programowania np.: C/C++ w systemie wbudowanym Debian Linux, • Obsługa układów PWM za pomocą instrukcji z terminala i wybranego języka programowania np.: C/C++ w systemie wbudowanym Debian Linux, • Obsługa magistrali CAN za pomocą instrukcji z terminala i wybranego języka programowania np.: C/C++ w systemie wbudowanym Debian Linux, • Obsługa kamery wideo w systemie wbudowanym Debian Linux przy pomocy skryptu BASH i wybranego języka programowania C/C++, • Komunikacja Serwer – Klient z wykorzystaniem protokołu UDP. • Komunikacja Serwer – Klient z wykorzystaniem protokołu TCP. • Zdalny pomiar i udostępnienie wyników pomiaru za pomocą wybranych aplikacji internetowych. • Pomiar czasu w systemie wbudowanym z wykorzystaniem układów zegara czasu rzeczywistego RTC. Realizacja aplikacji w języku C/C++. • Obsługa akcelerometru w systemie wbudowanym z wykorzystaniem. Realizacja aplikacji w języku C/C++. • Obsługa watchdoga w systemach wbudowanych. • Realizacja aplikacji telemetrycznej wykorzystującej protokół MQ Telemetry Transport (MQTT) do pomiaru wybranej wielkości fizycznej np.: temperatura, prąd itp. • Wykorzystanie środowiska programistycznego Qt do tworzenia aplikacji IoT z graficznym interfejsem użytkownika. • Wykorzystanie koprocessorów (PRU-ICSS) Programmable Real-Time Unit and Industrial Communication Subsystem w systemach wbudowanych.	Ćwiczenia laboratoryjne	U1, U2, K1

4. Metody prowadzenia zajęć, weryfikacji efektów uczenia się i warunki zaliczenia

Forma zajęć	
-------------	--

Wykład	Metody prowadzenia zajęć:	
	Wykład	
	Metody (sposoby) weryfikacji:	Udział:
	Zaliczenie pisemne	100%
	Warunki zaliczenia przedmiotu:	
	Wykład: Zaliczenie pisemne – zaliczenie od 51% wykonanych zadań.	
Ćwiczenia laboratoryjne	Metody prowadzenia zajęć:	
	Ćwiczenia laboratoryjne	
	Metody (sposoby) weryfikacji:	Udział:
	Sprawozdanie	100%
	Warunki zaliczenia przedmiotu:	
	Laboratorium: Wykonanie obowiązkowego zestawu ćwiczeń z użyciem zestawów laboratoryjnych oraz przygotowanie sprawozdań, 1 dodatkowe ćwiczenie ocena 4,0, 2 dodatkowe ćwiczenia ocena 5,0.	

Efekt uczenia się dla przedmiotu	Metody (sposoby) weryfikacji	
	Zaliczenie pisemne	Sprawozdanie
W1	x	
W2	x	
U1		x
U2		x
K1		x

5. Literatura

Literatura podstawowa

1. Derek Molloy, Exploring Beaglebone Tools and Techniques for building with embedded Linux, Willey 2019.
2. Derek Molloy, Exploring Raspberry Pi: Interfacing to the Real World with Embedded Linux. New York: Wiley, 978-1119188687, 720pp, 2016.
3. Marek Galewski, Aplikacje i ćwiczenia w języku C z biblioteką HAL Wydawnictwo BTC Legionowo 2019.
4. Robert Brzoza-Woch, Mikrokontrolery AT91SAM7 w przykładach , Wydawnictwo BTC Legionowo 2009.

Literatura uzupełniająca

1. Anderson, Rick, Arduino dla zaawansowanych, Helion 2014.
2. R. Love, „Linux programowanie systemowe”, Wydawnictwo O’Reilly, 2014.
3. Maciej Szumski, Mikrokontrolery STM32 w systemach sterowania i regulacji, Wydawnictwo BTC Legionowo 2017.

6. Nakład pracy studenta - bilans godzin i punktów ECTS

Aktywność studenta		Obciążenie studenta Liczba godzin
Zajęcia prowadzone z bezpośrednim udziałem nauczyciela akademickiego lub innych osób prowadzących zajęcia	Wykład	30
	Ćwiczenia laboratoryjne	30
Praca własna studenta	Konsultacje	2
	Przygotowanie do zajęć	8
	Studiowanie literatury	10
	Przygotowanie do zaliczenia	20
Łączny nakład pracy studenta		100
Liczba punktów ECTS		4

* Godzina (dydaktyczna) oznacza 45 minut