



**POLITECHNIKA
BYDGOSKA**

Wydział Telekomunikacji,
Informatyki i Elektrotechniki

Karta przedmiotu
Programowanie i konfiguracja systemów wbudowanych

1. Informacje podstawowe

Kierunek studiów elektronika i telekomunikacja Specjalność: sieci teleinformatyczne Jednostka zarządzająca kierunkiem studiów Wydział Telekomunikacji, Informatyki i Elektrotechniki Poziom studiów drugiego stopnia (mgr inż.) Profil studiów Profil ogólnoakademicki Forma studiów studia niestacjonarne		Cykl kształcenia (nabór) 2024/25 Kod przedmiotu 05EITSTELN.DI4D.0372.24 Języki wykładowe polski Obligatoryjność Obligatoryjny specjalnościowy Blok zajęciowy Przedmioty specjalnościowe	
Wymagania wstępne	Podstawy programowania w języku wysokopoziomowych, znajomość podstawowych elementów logicznych, pamięciowych techniki cyfrowej, Budowa oraz działanie mikroprocesora.		
Przedmioty wprowadzające	Układy cyfrowe, Technika mikroprocesorowa		
Koordynator	Łukasz Saganowski		
Okres Semestr 3	Forma zaliczenia Egzamin Forma prowadzenia i godziny zajęć Wykład: 9 Ćwiczenia laboratoryjne: 18		Liczba punktów ECTS 3.0

2. Efekty uczenia się dla przedmiotu

Kod	Opis efektów uczenia się	Odniesienie do kierunkowych efektów uczenia się	Odniesienie do charakterystyk PRK
Wiedza:			
W1	Posiada wiedzę na temat budowy, zasady działania, projektowania, testowania oraz pomiarów systemów wbudowanych.	EIT_O2_K_W15	P7S_WG P7S_WG_inż
W2	Posiada wiedzę na temat zasad projektowania systemów wbudowanych z wykorzystaniem języków programowania zorientowanych obiektowo np. języka C++.	EIT_O2_K_W18	P7S_WG P7S_WG_inż
Umiejętności:			
U1	Potrafi wykorzystywać wiedzę z zakresu elektroniki, telekomunikacji oraz systemu wbudowanego Linux do rozwiązywania rzeczywistych problemów inżynierskich.	EIT_O2_K_U20	P7S_UW P7S_UW_inż
U2	Potrafi znaleźć nowe zastosowania dla implementacji systemów wbudowanych opartych o system Linux do realizacji rzeczywistych i użytecznych celów. Potrafi zastępować istniejące rozwiązania techniczne z wykorzystaniem systemów wbudowanych w celu poprawy ich funkcjonalności i efektywności.	EIT_O2_K_U27	P7S_UW P7S_UW_inż
Kompetencje społeczne:			
K1	Potrafi wykorzystywać zdobytą wiedzę oraz umiejętności z zakresu systemów wbudowanych w celu opracowywania użytecznych społecznie rozwiązań technicznych oraz ma świadomość ciągłej potrzeby wdrażania innowacyjnych rozwiązań technicznych.	EIT_O2_K_K03	P7S_KK

3. Treści programowe

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
1.	Wykład obejmuje następujące zagadnienia: □ Budowa sprzętowa wybranych systemów wbudowanych opartych o np. proste mikrokontrolery RISC typu AVR, procesory typu ARM serii ARM7, CortexMx oraz o zaawansowane układy typu SoC System on Chip. □ Przegląd typów systemów operacyjnych stosowanych w systemach wbudowanych. □ Instalacja oraz konfiguracja wybranych systemów wbudowanych opartych o np. prosty system wbudowany oparty o biblioteki Arduino, system wbudowany oparty o biblioteki Nut/OS. □ Instalacja oraz konfiguracja wybranych systemów wbudowanych opartych o np. system wbudowany oparty o wbudowany system operacyjny Linux np. Debian. □ Realizacja wybranych projektów (podłączanie czujników, oprogramowania stosu TCP/IP, obsługa przerwań, przetwornik ADC, obsługa układów czasowych itp.) w oparciu o podstawowy system wbudowany Arduino. □ Charakterystyka wbudowanych systemów operacyjnych Linux opartych na przykładzie dystrybucji Debian Linux. □ Budowa sprzętowa wbudowanego systemu operacyjnego działającego pod kontrolą Linux na podstawie układu opartego o SoC np.: Beagle Bone Black. □ Konfiguracja środowiska sieciowego. □ Proces bootowania systemu Linux na podstawie wybranego systemu wbudowanego np. BBB - Beagle Bone Black, Pocket Beagle. □ 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, oraz magistral szeregowych I2C, SPI, UART w systemie wbudowanym z wykorzystaniem sterowników wbudowanych w Linux. □ Sposoby obsługi urządzeń w systemie wbudowanym Linux. □ Kompilacja skrośna (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. □ Tworzenie aplikacji wielowątkowych dla systemu wbudowanego Linux. □ Elementy programowania systemowego w Linux. □ Obsługa watchdog w systemie wbudowanym Linux. □ Aplikacje klient-serwer współpracujące z chmurą. STRONA 81 z 113 WTiE EiT 2019/2020 Załącznik nr 3 do: Wytycznych do projektowania i modyfikacji programów studiów I i II stopnia w UTP Projektowanie obwodów drukowanych dla systemów wbudowanych. □ Podłączanie elementów wykonawczych takich jak: przełączniki, przełączniki półprzewodnikowe, silniki i inne elementy mocy. □ Pomiar wielkości elektrycznych (np.: bezkontaktowy pomiar prądu za pomocą sensora Halla, napięcia) i nieelektrycznych (np.: natężenie światła, chromatyczność, odległość itp.) w systemach wbudowanych.	Wykład	W1, W2

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
2.	Ćwiczenia laboratoryjne obejmują następujące zagadnienia: 1. Instalacja i konfiguracja narzędzi programistycznych dla systemu wbudowanego opartego o biblioteki Arduino. 2. Obsługa portów GPIO dla systemu Arduino. 3. Obsługa układów czasowych, przetwornika ADC, obsługa modulacji PWM. 4. Obsługa przerwań dla środowiska programistycznego Arduino. 5. Obsługa stosu TCP/IP dla układów opartych o układy typu AVR. 6. Projektowanie wybranych serwisów internetowych np. serwera HTTP wraz z prezentacją oraz interakcją użytkownika. 7. Projektowanie sensora sieciowego przesyłającego wyniki pomiarów (np. temperatury) do bazy danych. 8. Instalacja oraz konfiguracja systemu wbudowanego Debian Linux w oparciu o platformę Beagle Bone Black, PocketBeagle. 9. Obsługa portu GPIO za pomocą instrukcji z terminala i skryptu BASH w systemie Debian Linux - BBB, 10. Obsługa magistrali szeregowej I2C za pomocą instrukcji z terminala i wybranego języka programowania np.: C/C++ w systemie wbudowanym Debian Linux. 11. Obsługa przetwornika ADC za pomocą instrukcji z terminala i wybranego języka programowania np.: C/C++ w systemie wbudowanym Debian Linux. 12. Obsługa kamery wideo w systemie wbudowanym Debian Linux przy pomocy skryptu BASH i wybranego języka programowania C/C++. 13. Obsługa akcelerometru w systemie wbudowanym Debian Linux. 14. Projektowanie aplikacji typu klient - serwer dla wybranego zastosowania w systemie Debian Linux. 15. Projektowanie aplikacji dla systemu wbudowanego Linux współpracującego z chmurą. 16. Wykorzystanie watchdoga w systemie wbudowanym Linux. 17. Projektowanie sterowników urządzeń dla systemu wbudowanego Debian Linux. 18. Wykorzystanie środowiska programistycznego Qt do tworzenia aplikacji IoT.	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: Egzamin pisemny – zaliczenie od 51% wykonanych zadań.	

Ćwiczenia laboratoryjne	Metody prowadzenia zajęć:	
	Ćwiczenia laboratoryjne	
	Metody (sposoby) weryfikacji:	Udział:
	Sprawozdanie	100%
	Warunki zaliczenia przedmiotu:	
	Wykonanie obowiązkowego zestawu ćwiczeń z użyciem zestawu laboratoryjnego 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. Love R., Linux programowanie systemowe, Wydawnictwo O'Reilly, 2014
2. Molloy D., Exploring Beaglebone Tools and Techniques for building with embedded Linux, Willey, 2019.

Literatura uzupełniająca

1. Doliński J., Mikrokontrolery AVR w praktyce, BTC, 2004
2. Anderson R., Arduino dla zaawansowanych, Helion, 2014

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	9
	Ćwiczenia laboratoryjne	18
Praca własna studenta	Konsultacje	2
	Przygotowanie do zajęć	14
	Studiowanie literatury	10
	Przygotowanie do egzaminu	22

Łączny nakład pracy studenta	75
Liczba punktów ECTS	3

* Godzina (dydaktyczna) oznacza 45 minut