POLITECHNIKA GDAŃSKA

Karta przedmiotu

Nazwa i kod przedmiotu	Programowanie mikrokontrolerów, PG_00072352		
Kierunek studiów	Technologie Przemysłu 5.0		
Data rozpoczęcia studiów	październik 2024 r.	Rok akademicki realizacji przedmiotu	2026/2027
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć	Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki
Forma studiów	stacjonarne	Sposób realizacji	na uczelni
Rok studiów	3	Język wykładowy	polski
Semestr studiów	5	Liczba punktów ECTS	4.0
Profil kształcenia	ogólnoakademicki	Forma zaliczenia	egzamin
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Inżynierii Biomedycznej		
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	dr inż. Grzegorz Jasiński	
	Prowadzący zajęcia z przedmiotu	inż. Mateusz Golan dr inż. Grzegorz Jasiński	

Formy zajęć							
	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	30.0	0.0	0.0	45
	W tym liczba godzin zajęć na odległość: 0.0						

Aktywność studenta i liczba godzin pracy					
	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach	Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	45	3.0	52.0	100

Cel przedmiotu	
	Celem przedmiotu jest zdobycie wiedzy i praktycznych umiejętności programowania mikrokontrolerów (AVR, STM32) w kontekście systemów wbudowanych Przemysłu 5.0. Student poznaje architekturę układów oraz zasady konfiguracji peryferiów: portów GPIO, układów zegarowych, timerów, RTC, systemu przerwań (IRQ), mechanizmu DMA i interfejsów komunikacyjnych (m.in. UART, I2C). Zajęcia przygotowują do projektowania, implementacji i uruchamiania oprogramowania niskopoziomowego dla inteligentnych urządzeń sterujących, pomiarowych oraz systemów interakcji człowiek–maszyna.

Efekty uczenia się przedmiotu

Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
[K6_K02] samodzielnie podejmuje decyzje, przeprowadza krytyczną ocenę działań własnych oraz działań zespołów, którymi kieruje, jest gotów do podejmowania decyzji i przyjmowania odpowiedzialności za skutki tych działań	Jest gotów do samodzielnego podejmowania decyzji inżynierskich przy tworzeniu oprogramowania niskopoziomowego (AVR, STM32) oraz przyjmowania odpowiedzialności za techniczne, energetyczne i eksploatacyjne skutki jego działania w urządzeniach sterujących i pomiarowych.	[SK1] Ocena umiejętności pracy w grupie [SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce
[K6_W02] wykazuje się znajomością i zrozumieniem elektroniki, automatyki oraz telekomunikacji a także teorii systemów, pozwalającą identyfikować problemy i formułować rozwiązania odpowiednie dla czwartej i piątej rewolucji przemysłowej	Zna i rozumie architekturę mikrokontrolerów (AVR, STM32) oraz zasady działania ich układów peryferyjnych (GPIO, układy zegarowe, timery, RTC, system przerwań IRQ, DMA) i interfejsów komunikacyjnych (m.in. UART, I2C) w powiązaniu z podstawami elektroniki, automatyki, telekomunikacji i teorii systemów, co pozwala identyfikować problemy i formułować rozwiązania wbudowane dla Przemysłu 4.0 i 5.0.	[SW1] Ocena wiedzy faktograficznej [SW2] Ocena wiedzy zawartej w prezentacji [SW3] Ocena wiedzy zawartej w opracowaniu tekstowym i projektowym
[K6_U02] identyfikuje i rozwiązuje problemy związane z przetwarzaniem i transmisją sygnałów, integrować systemy pomiarowe z systemami sterowania i zarządzać systemami elektronicznymi w kontekście inteligentnych procesów produkcyjnych	Potrafi identyfikować i rozwiązywać problemy związane z przetwarzaniem i transmisją sygnałów w mikrokontrolerach (AVR, STM32), integrować układy pomiarowe z układami sterowania (z wykorzystaniem GPIO, timerów, RTC, przerwań IRQ, DMA oraz interfejsów UART i I2C) oraz zarządzać pracą systemów elektronicznych w kontekście inteligentnych procesów produkcyjnych.	[SU1] Ocena realizacji zadania [SU4] Ocena umiejętności korzystania z metod i narzędzi [SU3] Ocena umiejętności wykorzystania wiedzy uzyskanej w ramach przedmiotu

Treści przedmiotu

Treści przedmiotu - wykład

Architektura mikrokontrolerów ze szczególnym uwzględnieniem 8-bitowych układów AVR (architektura harwardzka, organizacja pamięci Flash, SRAM i EEPROM, rejestry robocze oraz SFR) z wprowadzeniem do 32-bitowej architektury STM32 (ARM Cortex-M). Programowanie niskopoziomowe w języku C: operacje bitowe, maski, dostęp rejestrowy, modyfikator volatile oraz podstawy warstw abstrakcji sprzętu. Układy zegarowe (clocks): źródła taktowania, *fuse-bity* i preskalery w AVR, konfiguracja drzewa zegarów w STM32 oraz tryby energooszczędne.

Porty wejścia/wyjścia (GPIO): konfiguracja rejestrów (DDR, PORT, PIN), stany elektryczne i rezystory podciągające, eliminacja drgań styków oraz obsługa elementów interfejsu użytkownika (LED, przyciski, wyświetlacze). System przerwań (IRQ): tablica wektorów przerwań AVR, przerwania zewnętrzne i peryferyjne, sekcje krytyczne, determinizm czasowy oraz kontroler NVIC w STM32. Układy licznikowo-czasowe (timery): tryby pracy (Normal, CTC, PWM), sterowanie układami wykonawczymi oraz obsługa zegara czasu rzeczywistego (RTC – zewnętrznego dla AVR i wbudowanego w STM32).

Interfejsy komunikacyjne: asynchroniczny UART/USART (ramkowanie, obsługa w przerwaniach, bufory kołowe) oraz synchroniczna magistrala I2C/TWI (adresowanie, komunikacja z czujnikami, RTC i pamięcią EEPROM). Bezpośredni dostęp do pamięci (DMA) w układach STM32 jako metoda odciążania procesora przy transmisji i przetwarzaniu danych. Podstawy przetwarzania analogowo-cyfrowego (ADC), układy nadzorujące (Watchdog, BOR) oraz diagnostyka systemów wbudowanych w zastosowaniach Przemysłu 5.0.

Treści przedmiotu - laboratoria

Konfiguracja środowiska i programatora, podstawy taktowania mikrokontrolera AVR, operacje bitowe na rejestrach kierunkowych i portowych, sterowanie diodami LED, obsługa przycisków z programową eliminacją drgań styków oraz wyświetlacza LCD 2×16.
Konfiguracja timerów sprzętowych, obsługa przerwań licznikowych oraz przerwań zewnętrznych, realizacja nieblokującego harmonogramowania zadań i dynamicznego multipleksowania wyświetlacza 7-segmentowego.
Sprzętowe generowanie przebiegów PWM z wykorzystaniem timerów do płynnej regulacji mocy i prędkości, z uwzględnieniem podstaw pomiaru sygnałów analogowych.
Konfiguracja interfejsu USART i transmisji z komputerem PC, obsługa transmisji w przerwaniach z wykorzystaniem buforów kołowych oraz implementacja prostego protokołu poleceń diagnostycznych i sterujących.
Magistrala I2C (TWI) i obsługa zewnętrznego zegara czasu rzeczywistego.
Wprowadzenie do 32-bitowej architektury STM32.
Kontroler DMA, komunikacja (UART/I2C) oraz wewnętrzny RTC i tryby energooszczędne.

Wymagania wstępne i dodatkowe	Znajomość podstaw programowania w języku C (typy danych, wskaźniki, struktury, operacje bitowe), podstaw elektroniki i techniki cyfrowej (systemy liczbowe: binarny i szesnastkowy, logika cyfrowa, układy kombinacyjne i sekwencyjne, podstawy obwodów elektrycznych) oraz podstaw architektury komputerów.	
Sposoby i kryteria oceniania osiąganych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy / Składowa oceny końcowej
	laboratorium	50.0% / 35.0%
	wykład	50.0% / 65.0%
Zalecana lista lektur	Podstawowa lista lektur	• Kardaś M., Mikrokontrolery AVR. Język C – podstawy programowania, Wydawnictwo Atnel, Szczecin (wyd. II i kolejne). (Pozycja bezpośrednio powiązana sprzętowo z zestawem uruchomieniowym ATB 1.05a). • Francuz T., Język C dla mikrokontrolerów AVR. Od podstaw do zaawansowanych aplikacji, Wydawnictwo Helion, Gliwice, 2015. • Galewski M., STM32. Aplikacje i ćwiczenia w języku C z biblioteką HAL, Wydawnictwo BTC, Legionowo, 2017. • Microchip Technology Inc., ATmega32A – 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash. Data Sheet, dokumentacja producenta (wersja aktualna online). • STMicroelectronics, RM0351 Reference Manual: STM32L47xxx, STM32L48xxx, STM32L49xxx and STM32L4Axxx advanced Arm-based 32-bit MCUs oraz STM32L496xx Datasheet, dokumentacja producenta (wersja aktualna online).
	Uzupełniająca lista lektur	• Kardaś M., Język C. Pasja programowania mikrokontrolerów 8-bitowych, Wydawnictwo Atnel, Szczecin, 2014. (Rozszerzenie m.in. o zaawansowaną obsługę przerwań, I2C, RTC i budowę modułowego kodu na zestawie ATB). • Noviello C., Mastering STM32, wyd. II, Leanpub, 2022. (Szczegółowe omówienie drzewa zegarów RCC, kontrolera NVIC, DMA, UART, I2C, RTC oraz trybów Low-Power). • Yiu J., The Definitive Guide to ARM Cortex-M3 and Cortex-M4 Processors, wyd. III, Newnes / Elsevier, 2013. • Williams E., AVR Programming: Learning to Write Software for Hardware, Maker Media / O'Reilly, 2014. • Dokumentacja techniczna zestawów laboratoryjnych: Atnel, Instrukcja użytkownika i schemat ideowy zestawu uruchomieniowego ATB 1.05a. Kamami, Dokumentacja techniczna zestawu uruchomieniowego KAmeleon-STM32L4 (STM32L496ZGT6).
	Adresy eZasobów	

Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	Zagadnienia wykład: • Porównanie 8-bitowej architektury harwardzkiej AVR (organizacja pamięci Flash, SRAM, EEPROM, rejestry SFR) z 32-bitową architekturą ARM Cortex-M (STM32). • Znaczenie operacji bitowych, masek oraz modyfikatora volatile w programowaniu niskopoziomowym w języku C i dostępie do rejestrów GPIO (DDR, PORT, PIN). • Konfiguracja układów zegarowych: źródła taktowania, fuse-bity i preskalery w AVR oraz drzewo zegarów (RCC) w STM32 w kontekście zużycia energii. • Mechanizm obsługi przerwań (IRQ) w AVR oraz kontroler NVIC/EXTI w STM32 – tablica wektorów, priorytety, sekcje krytyczne, determinizm czasowy i zasady pisania krótkich procedur ISR. • Tryby pracy timerów oraz wyznaczanie wartości preskalera i rejestrów porównania dla zadanej częstotliwości i współczynnika wypełnienia. • Zasada transmisji asynchronicznej UART/USART (ramkowanie, obsługa w przerwaniach, bufory kołowe) oraz synchronicznej magistrali I2C/TWI (warunki START/STOP, adresowanie, potwierdzenia ACK/NACK). • Rola kontrolera DMA, wewnętrznego/zewnętrznego zegara RTC, układów nadzorujących (Watchdog, BOR) oraz trybów obniżonego poboru mocy w niezawodnych systemach Przemysłu 5.0 Przykładowe realizowane zadania na laboratorium: • Implementacja nieblokującej obsługi przycisków (z eliminacją drgań styków) oraz sterowania diodami LED i wyświetlaczem LCD/7-segmentowym. • Konfiguracja przerwań licznikowych oraz sprzętowego sygnału PWM do regulacji jasności lub prędkości układu wykonawczego (np. sterownika silnika L293D) w mikrokontrolerze AVR. • Budowa dwukierunkowego łącza komunikacyjnego UART (AVR–PC) z wykorzystaniem przerwań i buforów kołowych do przesyłania poleceń sterujących i danych diagnostycznych. • Odczyt i zapis czasu w zewnętrznym układzie RTC oraz danych w pamięci EEPROM po magistrali I2C (TWI) z wizualizacją na wyświetlaczu.
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.