

SYLABUS
DOTYCZY CYKLU KSZTAŁCENIA 2023-2027
 Rok akademicki 2025/2026

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE/MODULE

Nazwa przedmiotu	<i>inżynieria systemów mikroinformatycznych</i>
Kod przedmiotu	
Nazwa jednostki prowadzącej kierunek	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>informatyka</i>
Poziom studiów	<i>studia inżynierskie I-go stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr studiów	<i>rok III, semestr 6</i>
Rodzaj przedmiotu	<i>przedmiot kierunkowy</i>
Język wykładowy	<i>polski</i>
Koordynator	<i>dr inż. Bogusław Twaróg</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Bogusław Twaróg</i>

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt ECTS
6	12			13					2

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Student posiada umiejętność korzystania ze sprzętu komputerowego w ramach użytkowania profesjonalnego oprogramowania inżynierskiego. Posiada przygotowanie w zakresie budowy, zasady działania oraz podstaw programowania komputerów. Zna podstawy fizyki, techniki cyfrowej, systemów operacyjnych.

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE**3.1 Cele przedmiotu**

C ₁	Przedstawienie zasad działania i programowania systemów mikroprocesorowego sterowania z wykorzystaniem produktów wybranych firm.
C ₂	Zapoznanie z zasadami projektowania i uruchamiania prostych układów automatyki opartych na jednoukładowych systemach mikroprocesorowych.
C ₃	Utrwalanie umiejętności programistycznych w korelacji z modelowaniem i implementowaniem w praktycznych układach sterujących.

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu Student:	Odniesienie do efektów kierunkowych
EK_01	Zna techniki informatyczne i sprzęt komputerowy do wspomagania projektowania i zarządzania specjalizowanym oprogramowaniem inżynierskim. Posiada wiedzę potrzebną do modelowania, symulacji i syntezy systemów cyfrowych z wykorzystaniem języków opisu sprzętu.	K_Wo7
EK_02	Potrafi skonfigurować sprzęt i oprogramowanie w środowisku programowania praktycznego systemu sterowania. Potrafi opracowywać algorytmy sterowania dla podstawowych zadań automatyki, realizować programy sterowania i testować ich działanie.	K_Uo3
EK_03	Świadomie korzysta z wiarygodnych źródeł dotyczących omawianej tematyki i samodzielnie poddaje je krytycznej analizie.	K_Uo3

3.3 Treści programowe**A. Problematyka wykładu**

Omówienie tematyki przedmiotu, literatury, form i zasad zaliczenia.
Jednostki miar w elektrotechnice. Podstawowe pojęcia (ładunek, napięcie, prąd, moc, energia). Klasyfikacja i podział elementów, sygnałów i obwodów elektrycznych. Elementy obwodu. Łączenie szeregowo i równoległe elementów. Wyznaczanie rezystancji zastępczej obwodu. Prąd stały. Prawo Ohma i Prawa Kirchhoffa.
Diody półprzewodnikowe -budowa, właściwości, rodzaje, modele i zastosowania (prostowniki, stabilizatory). Elementy optoelektroniczne - technologie, rodzaje, obszary zastosowań i kierunki rozwoju. Tranzystory bipolarne- podstawowe właściwości, układy pracy, charakterystyki i układy polaryzacji.
Przetworniki A/C i C/A. Przegląd przetworników A/C. Parametry techniczne, metody poprawiania parametrów dokładności, przegląd przekształtników C/A. Sterowanie PWM. Komparatory analogowe.
Tendencje rozwojowe i przegląd układów mikroprocesorowych. Budowa i zasada działania systemu mikroprocesorowego. Mikrokomputery jednoukładowe (mikrokontrolery) – charakterystyka. Architektura 8-bitowych mikrokontrolerów rodziny ATmega firmy Atmel.
Dobór i programowanie układów wejścia / wyjścia. Typowe układy wejściowe, stosowane w

prostych mikrokomputerach budowanych do celów przemysłowych: bramki, przetworniki A/C, klawiatury. Typowe układy wyjściowe: zatraski, przetworniki C/A, wyświetlacze diodowe i wyświetlacze LCD alfanumeryczne i graficzne, wyjścia PWM.
Implementacja algorytmów przetwarzania sygnałów na mikrokontrolerach ATmega. Uruchamianie i testowanie algorytmów w wybranych środowiskach programistycznych na przykładach stosowanych w urządzeniach sterowania przemysłowego.
Zastosowanie metod programowania strukturalnego i obiektowego w rozwiązywaniu zagadnień sterowania układami cyfrowymi i analogowymi dla podstawowych rozwiązań architektur mikroprocesorowych.
Czytanie i wyświetlanie informacji w systemach mikroprocesorowych. Konstrukcja i zasady czytania klawiatur, sterowniki wyświetlaczy LED, LCD (alfanumerycznych i graficznych).
Powtórzenie materiału. Zaliczenie przedmiotu.

B. Problematyka ćwiczeń laboratoryjnych

Omówienie tematyki przedmiotu, literatury, form i zasad zaliczenia.
Zapoznanie z zestawami uruchomieniowymi platformy Arduino i oprogramowaniem narzędziowym oraz uruchamianie prostych programów w języku C, realizujących uwarunkowane czasowo sterowania binarne.
Wykorzystanie biblioteki standardowej. Metody obiektowe w programowaniu mikrokontrolerów. Efektywne projektowanie programów sterujących. Projektowanie i implementowanie aplikacji sterujących z zastosowaniem podstawowych czujników i aktuatorów cyfrowych.
Komunikacja z urządzeniami (IO pins, protocol pins). Pomiar częstotliwości, liczby impulsów, czasu trwania impulsu z wykorzystaniem liczników wewnętrznych i komparatora napięcia. Pomiar napięcia przy pomocy przetwornika A/C.
Obsługa programowa klawiatur sekwencyjnych i matrycowych. Modelowanie aplikacji w języku C i uruchomienie na platformie Arduino driver'a klawiatury.
Obsługa programowa wyświetlaczy alfanumerycznych typu LED i typu LCD. Projektowanie i implementowanie algorytmów w języku C i uruchomienie na platformie Arduino driver'a wyświetlacza alfanumerycznego LCD.
Mechanizm przerwań. Zapisywanie procedur obsługi przerwań. Przerwania generowane przez sygnały zewnętrzne oraz z układów wewnętrznych. Hierarchia przerwań. Włączanie i wyłączenie systemu obsługi przerwań.
Powtórzenie materiału. Zaliczenie przedmiotu.

3.4 Metody dydaktyczne

wykład z prezentacją multimedialną

ćwiczenia laboratoryjne: rozwiązywanie zadań, projekt.

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
EK_01	Rozmowa zaliczeniowa	w
EK_02	Sprawozdanie, obserwacja w trakcie zajęć	lab

EK_03	Sprawozdanie, obserwacja w trakcie zajęć	lab
-------	--	-----

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Laboratorium – zaliczenie, weryfikacja wiedzy i umiejętności odbywa się poprzez:

- odpytywanie studentów,
- sprawozdania tematyczne,
- ocenę pozytywną student uzyskuje w przypadku uzyskania minimum połowy możliwych do uzyskania punktów. Kolejne oceny równomiernie pokrywają skalę punktową.

Wykład

- rozmowa zaliczeniowa.

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	25
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	3
Godziny nie kontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	25
SUMA GODZIN	53
SUMARYCZNA LICZBA PUNKTÓW ECTS	2

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa:

1. Jakubiec J., „Podstawy techniki mikroprocesorowej”; Wyd. Politechniki Śląskiej, 2014
2. Audrey O'Shea, Elektronika i internet rzeczy. Przewodnik dla ludzi z prawdziwą pasją, Helion, 2021
3. R. Baranowski „Mikrokontrolery AVR ATmega w praktyce”; BTC 2005
4. Rick Anderson, Dan Cervo „Arduino dla zaawansowanych”, Helion 2014
5. Materiały udostępnione przez prowadzącego

Literatura uzupełniająca:

1. A. Pawlaczyk „Sztuka programowania mikrokontrolerów AVR – przykłady”; BTC 2007
2. Michael Margolis “Arduino Cookbook. 2nd Edition”, 2011 r.
3. Tomasz Francuz “Język C dla mikrokontrolerów AVR. Od podstaw do zaawansowanych aplikacji”, Helion 2015 r.
4. Materiały udostępnione przez prowadzącego