



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Sterowniki mikroprocesorowe i ich programowanie [S1Mech2>SMiP1]

Przedmiot

Kierunek studiów
Mechatronika

Rok/Semestr
2/4

Studia w zakresie (specjalność)
–

Profil studiów
ogólnoakademicki

Poziom studiów
pierwszego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład	Laboratorium	Inne
15	15	0
Ćwiczenia	Projekty/seminaria	
0	0	

Liczba punktów ECTS

2,00

Koordynatorzy

mgr inż. Daniel Wyrwał
daniel.wyrwal@put.poznan.pl

Wykładowcy

Wymagania wstępne

Podstawowa wiedza z zakresu automatyki, informatyki, elektroniki, techniki układów cyfrowych oraz podstaw programowania w języku C.

Cel przedmiotu

Zapoznanie studentów z budową, działaniem i programowaniem prostych mikrokontrolerów, projektowaniem oraz programowaniem sterowników mikroprocesorowych, w tym układów sterowania urządzeniami mechatronicznymi z wykorzystaniem bibliotek Arduino.

Przedmiotowe efekty uczenia się

Wiedza:

- Znajomość podstawowej budowy i działania mikrokontrolerów oraz sterowników mikroprocesorowych, ich możliwości i parametrów technicznych.
- Obsługa portów, przetworników ADC, liczników, układów czasowych oraz transmisji szeregowej wykorzystując biblioteki Arduino.
- Znajomość zasad obsługi przerwań oraz integracji mikrokontrolerów ze sterownikami wykorzystując biblioteki Arduino

Umiejętności:

- Projektowanie prostych układów sterowania opartych na mikrokontrolerach 8- i 32-bitowych.
- Programowanie mikrokontrolerów w języku C, w tym obsługa wejść/wyjść, przerwań oraz transmisji szeregowej z wykorzystaniem bibliotek arduino.
- Implementacja metod komunikacji między urządzeniami z wykorzystaniem bibliotek arduino.

Kompetencje społeczne:

- Rozumienie potrzeby ciągłego kształcenia i śledzenia rozwoju technologii.
- Świadomość znaczenia systemów mikroprocesorowych dla gospodarki i społeczeństwa.
- Umiejętność organizowania pracy zespołowej przy projektach systemów wbudowanych.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

- Zaliczenie pisemne: pytania otwarte i testowe sprawdzające wiedzę teoretyczną.
- Zaliczenie laboratoriów: zaliczenie sprawdzające umiejętność pisania programu i programowania mikrokontrolera.
- Zaliczenie projektów: ocena dokumentacji i implementacji projektów sterowników mikroprocesorowych.

Treści programowe

- Budowa i programowanie mikrokontrolerów oraz sterowników mikroprocesorowych.
- Porty, podłączanie elementów, konfiguracja i programowanie.
- Deklaracja zmiennych, instrukcje warunkowe i pętle w sterowaniu.
- Komunikacja szeregową (UART, SPI, I2C), podstawy programowania.
- Liczniki, układy czasowe i PWM w sterowaniu.
- Przetworniki ADC: zastosowanie i programowanie.
- Przerwania w mikrokontrolerach i sterownikach mikroprocesorowych.
- Projektowanie sterowników mikroprocesorowych dla systemów mechatronicznych.

Tematyka zajęć

- Wykłady i prezentacje teoretyczne.
- Laboratoria obejmujące programowanie mikrokontrolerów.
- Projekty w zakresie implementacji systemów sterowania.

Metody dydaktyczne

- Wykłady oraz prezentacje multimedialne.
- Laboratoria praktyczne z programowania mikrokontrolerów.
- Projekty grupowe i indywidualne.

Literatura

Podstawowa:

- Oficjalna dokumentacja Arduino - [Arduino.cc](http://arduino.cc)

Dokumentacja dostępna na stronie docs.arduino.cc zawiera oficjalne informacje dotyczące programowania w środowisku Arduino. Obejmuje szeroki zakres tematów, od podstaw składni języka C/C++ używanego w Arduino, po szczegółowe opisy funkcji, bibliotek oraz przykłady projektów.

Uzupełniająca:

- Mikrokontrolery AVR Język C. Podstawy programowania - Mirosław Kardaś
Książka stanowi wprowadzenie do programowania mikrokontrolerów AVR w języku C. Omawia podstawowe zagadnienia związane z konfiguracją mikrokontrolera, obsługą peryferiów oraz wykorzystaniem kompilatorów i środowisk programistycznych.
- Symfonia C++ - Jerzy Grębosz
Klasyczna książka wprowadzająca w język C++, wyjaśniająca zarówno podstawy, jak i zaawansowane koncepcje programowania obiektowego. Jest szczególnie przydatna dla osób,

które chcą rozwinąć swoje umiejętności w zakresie programowania niskopoziomowego oraz zrozumieć różnice między C i C++.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	50	2,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	1,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	20	1,00