**Nazwa przedmiotu:**

Programowanie mikrokontrolerów w języku C

**Koordynator przedmiotu:**

Sławomir SZOSTAK

**Status przedmiotu:**

Fakultatywny ograniczonego wyboru

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Elektronika

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

PMIK

**Semestr nominalny:**

6 / rok ak. 2015/2016

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

60h zajęć + 25h (przygotowanie do realizacji projektu) + 10h (udział w konsultacjach) + 30h (przygotowanie dokumentacji wstępnej, prezentacji, dokumentacji końcowej) = 125h

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 15h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 30h |
| Projekt:  | 15h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

Zaliczenie przedmiotu Podstawy techniki mikroprocesorowej (TMIK)

**Limit liczby studentów:**

80

**Cel przedmiotu:**

Celem przedmiotu jest pogłębienie umiejętności programowania mikrokontrolerów w języku C, jak również pokazanie możliwości łączenia języka C i asemblera oraz zapoznanie studentów
z metodami projektowania i uruchamiania oprogramowania systemów cyfrowych.

**Treści kształcenia:**

Treść wykładu:
Zastosowanie mikrokontrolerów (2h) - schemat blokowy systemu wbudowanego, architektury wybranych typów mikrokontrolerów – ich porównanie i analiza pod kątem wydajności programowania w języku C. Kryteria wyboru mikrokontrolera do wybranych aplikacji.
Konfigurowanie mikrokontrolerów (1h) - konfigurowanie zasobów wewnętrznych mikrokontrolera (liczniki, interfejsy komunikacyjne, przetworniki, itd.), komunikacja z urządzeniami zewnętrznymi.
Wprowadzenie do języka C (2h) - różnice w stosunku do standardu ANSI C, nowe typy zmiennych, przestrzenie danych w pamięci i ich deklaracje, definiowanie funkcji, przekazywanie parametrów do funkcji, obsługa przerwań.
Programowanie w C (3h) - dobór optymalnych typów danych, przerwania, priorytety przerwań, preprocesor, pliki nagłówkowe, makra, struktura projektu wieloplikowego, struktura programu, systemy operacyjne.
Środowiska programistyczne (1h) – konfiguracja i możliwości zintegrowanych narzędzi
do tworzenia, symulowania i analizy oprogramowania, opcje projektu (na przykładzie oprogramowanie µVision Keil i AVR Studio).
Techniki uruchamiania programu (1h) - debuggowanie programu przy użyciu symulatora programowego, symulacja urządzeń I/O, podglądanie zmiennych, rejestrów, zawartości pamięci, pułapki, praca krokowa, analiza pliku \*.m51, debuggowanie z wykorzystaniem systemów dostępnych w laboratorium.
Wybrane zagadnienia programowania (2h) - optymalizacja kodu, łączenie języka c i asemblera, uniezależnianie kodu od rozwiązań sprzętowych (częstotliwości zegara, liczby cykli maszynowych), autotestowanie oprogramowania, umieszczenie kodu wynikowego w wyznaczonych miejscach pamięci, techniki wykonywania modyfikacji oprogramowania.
Mikrokontrolery w zastosowaniach niskomocowych (1h), sposoby minimalizowania poboru mocy, dobór częstotliwości pracy układu, wykorzystywanie trybów energooszczędnych.
Przykładowe rozwiązania (2h) – obsługa wyświetlaczy LCD, bufor cyrkulacyjny do obsługi portu szeregowego, auto baud rate, przetwornik C/A z wykorzystaniem PWM, interfejs IIC.

Zakres laboratorium:
W ramach laboratorium student otrzymuje jedno zadanie polegające na napisaniu oprogramowania wielozadaniowego systemu mikroprocesorowego oraz uruchomieniu go np. w oparciu o bazę sprzętową laboratorium. Elementy oprogramowywanego systemu mikroprocesorowego, to m.in.: wyświetlacze LCD, LED, klawiatura, przetworniki analogowo-cyfrowe i cyfrowo-analogowe, komunikacja za pomocą magistrali I2C bus (EEPROM, zegar czasu rzeczywistego), komunikacja szeregowa, moduł wagowy, moduł GPS i GSM, moduł radiowy.

W ramach projektu studenci uzgadniają z prowadzącym sposób realizacji zadania oraz w dokonują krótkiej prezentacji przyjętych założeń, napotkanych problemów i sposobów ich rozwiązania.

**Metody oceny:**

zaliczenie projektu
ocena prezentacji
analiza dokumentacji wstępnej
analiza dokumentacji końcowej

**Egzamin:**

nie

**Literatura:**

Intel, "MCS 51 Microcontroller Family User's Manual"
Keil Software, “C51 Compiler – optimizing 8051 C Compiler and Library Reference”
Keil Software, “Macro Assembler and Utilities”
T. Starecki, “Mikrokontrolery jednoukładowe rodziny 51”, NOZOMI, Warszawa 1996.
J. Bogusz, „Programowanie mikrokontrolerów 8051 w języku C w praktyce”, BTC, 2005
Andrzej Witkowski, „Mikrokontrolery AVR programowanie w języku C przykłady zastosowań”, Pracownia Komputerowa Jacka Skalmierskiego, 2006
Tomasz Francuz, "Język C dla mikrokontrolerów AVR Od podstaw do zaawansowanych aplikacji", Helion, 2011
Krzysztof Paprocki, "Mikrokontrolery STM32 w praktyce", BTC, 2011

**Witryna www przedmiotu:**

http://www.pmik.imio.pw.edu.pl

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt PMIK\_W01:**

Zna aktualne trendy rozwoju mikrokontrolerów

Weryfikacja:

zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_W13

**Powiązane efekty obszarowe:** T1A\_W05

**Efekt PMIK\_W02:**

Zna kryteria i metodykę doboru architektury mikrokontrolera do konkretnego zastosowania

Weryfikacja:

zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_W09, K\_W12, K\_W15

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04, T1A\_W07, T1A\_W04, T1A\_W07, T1A\_W08

**Efekt PMIK\_W03:**

Zna zasady tworzenia oprogramowania dla systemów mikroprocesorowych

Weryfikacja:

zaliczenie projektu

**Powiązane efekty kierunkowe:**

**Powiązane efekty obszarowe:**

**Efekt PMIK\_W04:**

Zna podstawowe techniki uruchamiania oprogramowania

Weryfikacja:

zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_W04

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt PMIK\_U01:**

potrafi przeanalizować potrzeby sprzętowe i programowe aplikacji oraz zaproponować architekturę mikrokontrolera do jej realizacji

Weryfikacja:

zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_U08, K\_U16, K\_U17, K\_U19

**Powiązane efekty obszarowe:** T1A\_U05, T1A\_U09, T1A\_U10, T1A\_U12, T1A\_U14, T1A\_U15, T1A\_U16, T1A\_U14, T1A\_U01

**Efekt PMIK\_U02:**

Potrafi skonfigurować i korzystać ze środowiska programistycznego dedykowanego dla wybranego mikrokontrolera

Weryfikacja:

zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_U04, K\_U08, K\_U11, K\_U19

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U05, T1A\_U09, T1A\_U15, T1A\_U01

**Efekt PMIK\_U03:**

potrafi skonfigurować wybrane układy wewnętrzne mikrokontrolera (np. porty wejścia/wyjścia, liczniki, kontroler przerwań, UART, DMA) oraz napisać funkcje realizujące komunikację z wybranymi układami zewnętrznymi (np. klawiaturą, wyświetlaczem, przetwornikami A/C i C/A)

Weryfikacja:

zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_U04, K\_U08, K\_U16, K\_U17, K\_U18, K\_U19

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U05, T1A\_U09, T1A\_U10, T1A\_U12, T1A\_U14, T1A\_U15, T1A\_U16, T1A\_U14, T1A\_U09, T1A\_U16, T1A\_U01

**Efekt PMIK\_U04:**

Potrafi stworzyć strukturę projektu wieloplikowego poprzez podział zadania na odrębne bloki funkcjonalne i napisać oprogramowanie z wykorzystaniem różnego typu struktur danych i systemu przerwań mikrokontrolera

Weryfikacja:

zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_U04, K\_U08, K\_U18, K\_U19

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U05, T1A\_U09, T1A\_U16, T1A\_U01

**Efekt PMIK\_U05:**

potrafi napisać i uruchomić na wybranym systemie mikroprocesorowym oprogramowanie wykonujące ustalone z prowadzącym i zapisane w dokumentacji wstępnej projektu minimum funkcjonalności

Weryfikacja:

zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_U04, K\_U18, K\_U19

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U09, T1A\_U16, T1A\_U01

**Efekt PMIK\_U06:**

potrafi przygotować dokumentajcę wstępną projektu zawierającą opis funkcji realizowanych w projekcie, wykaz wykorzystywanych zasobów sprzętowych, harmonogram prac i ich podział w przypadku pracy zespołowej

Weryfikacja:

analiza dokumentacji wstępnej

**Powiązane efekty kierunkowe:** K\_U06, K\_U08

**Powiązane efekty obszarowe:** T1A\_U03, T1A\_U05

**Efekt PMIK\_U07:**

potrafi przygotować i wygłosić prezentację zawierającą opis wykonywanego projektu, stopień zaawansowania prac i dyskusję napotkanych problemów

Weryfikacja:

ocena prezentacji

**Powiązane efekty kierunkowe:** K\_U07, K\_U08, K\_U10

**Powiązane efekty obszarowe:** T1A\_U04, T1A\_U05, T1A\_U07

**Efekt PMIK\_U08:**

potrafi przygotować dokumentację końcową projektu zawierającą opis zrealizowanych funkcji, wykaz wykorzystywanych zasobów sprzętowych i algorytm działania programu

Weryfikacja:

analiza dokumentacji końcowej i zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_U06, K\_U10

**Powiązane efekty obszarowe:** T1A\_U03, T1A\_U07