Nazwa przedmiotu						
Programowanie systemów wbudowanych w języku C Programming of embedded systems in C language						
Dyscyplina					Oznaczenie przedmiotu	
Informatyka					1O_INFS2	
Rodzaj przedmiotu	Stopień studiów	Tryb studiów	Język zajęć		Rok	Semestr
do wyboru	2	stacjonarne	polski / angielski		I	II
Rodzaj zajęć	Wyk.	Ćw.	Lab.	Proj.	Sem.	Liczba punktów ECTS
Liczba godzin w semestrze	15	0	30	0	0	4
Koordynator	Dr hab. inż. Sławomir Gryś, prof. PCz. grys@el.pcz.czest.pl					
Prowadzący	Dr hab. inż. Sławomir Gryś, prof. PCz. grys@el.pcz.czest.pl Dr hab. inż. Stanisław Chudzik, prof. PCz. chudzik@el.pcz.czest.pl Asystent/Doktorant					

II. KARTA PRZEDMIOTU

Cel przedmiotu

C1. Nabycie umiejętności w zakresie zaawansowanego programowana w języku C systemów wbudowanych.
C2. Nabycie umiejętności używania zaawansowanych narzędzi do optymalizacji, testowania i walidacji kodu.

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1. Podstawowa wiedza z architektury komputerów, algorytmiki, podstaw programowania strukturalnego w językach wysokiego poziomu.
2. Umiejętność pracy samodzielnej i w grupie, w tym proponowania rozwiązania problemu technicznego.
3. Umiejętność korzystania ze źródeł literaturowych, specyfikacji technicznej.

Efekty kształcenia

EK1. Student posiada wiedzę i umiejętności w zakresie zaawansowanego programowana w języku C systemów wbudowanych.
EK2. Student posiada wiedzę i umiejętności w zakresie używania zaawansowanych narzędzi do optymalizacji, testowania i walidacji kodu.

Treści programowe: wykłady	Liczba godzin
W1 – Przegląd i porównanie architektur 8/16/32 bitowych procesorów przeznaczonych do systemów wbudowanych.	1,5
W2 – Komercyjne i open-source'owe środowiska uruchomieniowo-projektowe, walidatory kodu, profilery, bug tracking. Zarządzenie projektem - jak uniknąć bałaganu?	2
W3 – Podstawowa składnia języka ANSI C, rozszerzenia C90, C99, C11.	2
W4 – Programowanie w C z zachowaniem zgodności ze standardami MISRA, CERT_C, HIS.	1,5
W5 – Struktury danych: dynamiczne tablice, stosy, kolejki i drzewa.	1,5
W6 – Zaawansowane aspekty używania funkcji, funkcje wyższego rzędu.	2
W7 – Funkcje wariadyczne	1,5
W8 – Techniki programowania obiektowego w C	2
W9 – Test zaliczeniowy	1
SUMA	**15**

Treści programowe: laboratorium (ćwiczenia komputerowe)	Liczba godzin
L1 – Przedstawienie zasad odbywania zajęć, BHP oraz zasad zaliczenia laboratorium	1
L2 – Komercyjne i open-source'owe środowiska uruchomieniowo-projektowe, walidatory kodu, profilery, bug tracking. Zarządzenie projektem - jak uniknąć bałaganu?	4
L3 – Składnia języka ANSI C a rozszerzenia C90, C99, C11 na przykładach, tworzenie własnego programu.	5
L4 – Programowanie w C z zachowaniem zgodności ze standardami MISRA, CERT_C, HIS.	3
L5 – Struktury danych: dynamiczne tablice, stosy, kolejki i drzewa.	4
L6 – Zaawansowane aspekty używania funkcji, funkcje wyższego rzędu.	4
L7 – Funkcje wariadyczne	4
L8 – Techniki programowania obiektowego w C	3
L9 – Zaliczenie laboratorium/wpisy do indeksu	2
SUMA	**30**

Narzędzia dydaktyczne

1. Prezentacja multimedialna (wykład), programy demonstracyjne
2. Systemy uruchomieniowe z procesorami 8/16/32/64 bitowymi wraz z przygotowanymi przykładami

3. Komputery PC z zainstalowanym oprogramowaniem do programowania, optymalizacji, testowania i walidacji kodu.

Sposoby oceny efektów kształcenia (F – ocena Formująca, P – ocena Podsumowująca)

F1. Aktywność na wykładach (dyskusja, rozwiązywanie zagadnień przy tablicy).
F2. Aktywność podczas laboratorium.
P1. Zaliczenie na ocenę wykładu.
P2. Zaliczenie na ocenę zadań indywidualnych.

Obciążenie pracą Studenta

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe z prowadzącym	45
Zapoznanie się ze wskazaną literaturą	15
Przygotowanie do zajęć laboratoryjnych	15
Przygotowanie do zaliczenia wykładu	15
Sumaryczna liczba godzin/punktów ECTS dla przedmiotu	**90 / 4 ECTS**

Wykaz literatury podstawowej i uzupełniającej

1. Galewski M.: STM32. Aplikacje i ćwiczenia w języku C, Wyd. BTC, Legionowo 2011.
2. Sanchez J., Canton M.P.: "Embedded Systems Circuits and Programming, CRC Press, 2012.
3. Prata S.: Język C. Szkoła programowania, wyd. VI, Helion, Gliwice 2016.
3. Paprocki K.: Mikrokontrolery STM32 w praktyce, Wyd. BTC, Legionowo 2009.
2. Peter van der Linder: Expert C programming. Deep C secrets, Prentice Hall, NJ 1994.
3. Ball S.R.: Embedded Microprocessor Systems: Real World Design, Elsevier Science, 2002.
4. Borkowski P.: AVR i ARM7 Programowanie mikrokontrolerów dla każdego, Helion, Gliwice, 2010.
5. Francuz T.: Język C dla mikrokontrolerów AVR. Od podstaw do zaawansowanych aplikacji, Helion, Gliwice 2011.
8. Podręczniki (user's guide) środowisk programistycznych.

Macierz realizacji efektów kształcenia

Efekt kształcenia	Odniesienie efektu do efektów kształcenia dla dyscypliny naukowej Informatyka*	Cele przedmiotu	Forma zajęć	Narzędzia dydaktyczne	Sposób oceny
EK1	KINF2_W1, KIN2_U08	C1, C2	W, Lab	1, 2, 3	F1, P1
EK2	KINF2_W6, KINF2_U05	C3	W, Lab	1, 2, 3	F2, P2

* – wg załącznika

II. FORMY OCENY – SZCZEGÓŁY

Ocena	Efekty
EK1	**Student posiada wiedzę i umiejętności w zakresie zaawansowanego programowana w języku C systemów wbudowanych.**
2	Student nie posiada wiedzy ani umiejętności w zakresie programowana w języku C.
3	Student posiada elementarną wiedzę i umiejętności w zakresie programowana w języku C.
3.5	Student posiada elementarną wiedzę i umiejętności w zakresie programowana w języku C systemów wbudowanych.
4	Student posiada podstawową wiedzę i umiejętności w zakresie programowana w języku C systemów wbudowanych.
4.5	Student posiada podstawową wiedzę i umiejętności w zakresie zaawansowanego programowana w języku C systemów wbudowanych.
5	Student posiada wiedzę i umiejętności w zakresie zaawansowanego programowana w języku C systemów wbudowanych.
EK2	**Student posiada wiedzę i umiejętności w zakresie używania zaawansowanych narzędzi do optymalizacji, testowania i walidacji kodu.**
2	Student nie posiada wiedzy ani umiejętności w zakresie używania narzędzi do optymalizacji, testowania i walidacji kodu.
3	Student posiada wiedzę i umiejętności w zakresie używania narzędzi do optymalizacji, testowania i walidacji kodu.
3.5	Student posiada wiedzę i umiejętności w zakresie używania narzędzi do optymalizacji, testowania i walidacji kodu.
4	Student podstawową wiedzę i umiejętności w zakresie używania zaawansowanych narzędzi do optymalizacji, testowania i walidacji kodu.
4.5	Student posiada ugruntowaną wiedzę i umiejętności w zakresie używania zaawansowanych narzędzi do optymalizacji, testowania i walidacji kodu.
5	Student posiada wiedzę i umiejętności w zakresie używania zaawansowanych narzędzi do optymalizacji, testowania i walidacji kodu.

III. INNE PRZYDATNE INFORMACJE O PRZEDMIOCIE

1. Wszelkie informacje dla studentów na temat planu zajęć dostępne są na tablicy ogłoszeń oraz na stronie el.pcz.pl.
2. Prowadzący udostępnia na pierwszych zajęciach treści wykładów.
3. Informacje na temat warunków zaliczania zajęć przekazywana jest studentom podczas pierwszych zajęć.