Nazwa przedmiotu						
Systemy wbudowane Embedded systems						
Dyscyplina					Oznaczenie przedmiotu	
Informatyka					9K_INFS1	
Rodzaj przedmiotu	Stopień studiów	Tryb studiów	Język zajęć		Rok	Semestr
obowiązkowy	1	stacjonarne	polski / angielski		II	IV
Rodzaj zajęć	Wyk.	Ćw.	Lab.	Proj.	Sem.	Liczba punktów ECTS
Liczba godzin w semestrze	15	0	30	0	0	3
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. Poszerzenie wiedzy z zakresu mikrokontrolerów, języka C i Python, poznanie środowisk programistycznych.
C2. Nabycie umiejętności w zakresie projektowania układów wbudowanych pod kątem zastosowań przemysłowych.
C3. Nabycie umiejętności programowania mikrokontrolerów.

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji

1. Podstawowa wiedza z zakresu techniki cyfrowej, techniki mikroprocesorowej, algorytmiki, 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 zna działanie poszczególnych elementów systemu wbudowanego, w tym układów peryferyjnych mikrokontrolera oraz potrafi dobierać elementy i narzędzia pod kątem wymagań projektowych.
EK2. Student potrafi analizować, modyfikować oraz tworzyć oprogramowanie dla mikrokontrolerów integrując własny kod z funkcjami bibliotecznymi.

Treści programowe: wykłady	Liczba godzin
W1 – Systemy wbudowane – definicja, zastosowania. Przegląd i porównanie architektur uP 8/16/32 bitowych przeznaczonych do systemów wbudowanych. Architektura procesorów ARM, model programowy.	2
W2 – Komercyjne i open-source'owe środowiska uruchomieniowo-projektowe, narzędzia, programowanie mieszane, biblioteki, debugging, JTAG.	2
W3 – Składnia języka ANSI C.	2
W5 – Interfejsy szeregowe USART, SPI, 1Wire, I2C, USB.	2
W6 – Wykrywanie i korekcja błędów transmisji (bit parzystości, suma kontrolna, CRC).	2
W7 – Systemy czasu rzeczywistego. Dystrybucje Linuxa dla systemów wbudowanych.	2
W8 – Język Python, przetwarzanie skryptów.	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 – Instalacja i konfiguracja środowiska Arduino IDE, przykładowe aplikacje na system Intel Galileo 2	2
L3 – Realizacja indywidualnych zadań projektowych lub w zespołach dwuosobowych z wykorzystaniem zestawu urządzeń peryferyjnych (tzw. shields) typu: czujniki, moduł Bluetooth, moduł RF, moduł Wi-fi, moduł GPRS/GSM/GPS, sterownik silników krokowych i prądu stałego i in., praca na stanowiskach dydaktycznych.	4
L4 – Instalacja i konfiguracja środowiska Coocox, kompilatora GCC, wprowadzenie do tworzenia projektów dla systemu Red Bull na przykładzie sterowania diodą, praca z bibliotekami.	2
L5 – Operacje na liniach we/wy: brzęczek, przyciski, joystick, tworzenie własnej biblioteki.	2
L6 – Przetwarzanie A/C.	2
L7 – Obsługa wyświetlacza graficznego LCD i panelu dotykowego.	2
L8 – Konwersja grafiki rastrowej do kodu w C.	2
L9 – Układy czasowo-licznikowe, przerwania.	2
L10 – Jądro systemu czasu rzeczywistego, tworzenie i zarządzanie wątkami.	2
L11 – Instalacja Linuxa, protokół SSH, komendy Linuxa, transfer plików.	2

L12 – Wprowadzenie do Pythona	2
L13 – Odrabianie zajęć / realizacja indywidualnych projektów	4
L14– Zaliczenie laboratorium/wpisy do indeksu	1
SUMA	**30**

Narzędzia dydaktyczne

1. Prezentacja multimedialna (wykład), programy demonstracyjne
2. Systemy uruchomieniowe z procesorem ARM i Intel Quark wraz z przygotowanymi przykładami
3. Komputery PC z zainstalowanym oprogramowaniem: Coocox, Arduino IDE, dystrybucja Linuxa, kompilator GCC
4. Stanowiska dydaktyczne, urządzenia peryferyjne do współpracy z mikrokontrolerami

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 zadań wspólnych dla grupy.
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	10
Przygotowanie do zajęć laboratoryjnych	10
Przygotowanie do zaliczenia wykładu	10
Sumaryczna liczba godzin/punktów ECTS dla przedmiotu	**75 / 3 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. Paprocki K.: Mikrokontrolery STM32 w praktyce, Wyd. BTC, Legionowo 2009.
2. Augustyn J.: Projektowanie systemów wbudowanych na przykładzie rodziny SAM7S z rdzeniem ARM7TDMI, IGSMiE PAN, 2007.
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.
6. Chowdary Venkateswara Penumuchu: Simple Real-time Operating System. A Kernel Inside View for a Beginner, Trafford Publishing, Victoria (Kanada) 2007.
7. Bis M.: „Linux w systemach embedded", Wyd. BTC, Legiono 2011.
7. Specyfikacje techniczne mikroprocesorów, interfejsów szeregowych, urządzeń peryferyjnych.
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	KINF1_W12, KINF1_U05	C1, C2	W, Lab	1, 2, 3, 4	F1, P1
EK2	KINF1_W12, KINF1_U18	C3	Lab	2, 3	F2, P2

* – wg załącznika

II. FORMY OCENY – SZCZEGÓŁY

Ocena	Efekty
EK1	**Student zna działanie poszczególnych elementów systemu wbudowanego, w tym układów peryferyjnych mikrokontrolera oraz potrafi dobierać elementy i narzędzia pod kątem wymagań projektowych.**
2	Student nie zna działania elementów systemu wbudowanego, jego funkcji, ani podstawowych narzędzi.
3	Student zna działanie podstawowych elementów systemu wbudowanego, zna funkcje systemu wbudowanego, podstawowe narzędzia.
3.5	Student zna działanie podstawowych elementów systemu wbudowanego, zna funkcje systemu wbudowanego oraz potrafi dobierać podstawowe elementy i narzędzia.
4	Student zna działanie podstawowych elementów systemu wbudowanego, zna funkcje systemu wbudowanego oraz potrafi dobierać typowe elementy i narzędzia.
4.5	Student zna działanie elementów systemu wbudowanego, w tym układów peryferyjnych mikrokontrolera oraz potrafi dobierać większość elementów i narzędzi.
5	Student zna działanie elementów systemu wbudowanego, w tym układów peryferyjnych mikrokontrolera oraz potrafi dobierać elementy i narzędzia pod kątem wymagań projektowych.
EK2	**Student potrafi analizować, modyfikować oraz tworzyć oprogramowanie dla mikrokontrolerów integrując własny kod z funkcjami bibliotecznymi.**

2	Student nie potrafi analizować ani modyfikować ani tworzyć oprogramowania dla mikrokontrolerów.
3	Student korzystając z konsultacji potrafi analizować, modyfikować oraz tworzyć proste oprogramowanie dla mikrokontrolerów na podstawie przykładów i funkcji bibliotecznych.
3.5	Student w większości przypadków potrafi przeanalizować, modyfikować oraz stworzyć proste oprogramowanie dla mikrokontrolerów na podstawie przykładów, funkcji bibliotecznych.
4	Student potrafi samodzielnie analizować, modyfikować oraz tworzyć proste oprogramowanie dla mikrokontrolerów integrując własny kod z funkcjami bibliotecznymi.
4.5	Student potrafi samodzielnie analizować, modyfikować oraz tworzyć niezbyt złożone oprogramowanie dla mikrokontrolerów integrując własny kod z funkcjami bibliotecznymi.
5	Student potrafi samodzielnie przeanalizować, wyszukać, modyfikować oraz stworzyć oprogramowanie dla mikrokontrolerów wg założeń projektowych.

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.