



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie mikrokontrolerów jednoukładowych

Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Mikrosystemy informatyczne

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

2/3

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obieralny

Liczba godzin

Wykład

30

Ćwiczenia

Laboratoria

30

Projekty/seminaria

Inne (np. online)

Liczba punktów ECTS

4

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Mariusz Naumowicz

email: mnaumowicz@cs.put.poznan.pl

tel. +48 61 665-2364

Wydział Informatyki i Telekomunikacji

ul. Piotrowo 3 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Szymon Szczęsny

email: szymon.szczesny@cs.put.poznan.pl

tel. +48 61 665-2297

Wydział Informatyki i Telekomunikacji

ul. Piotrowo 3 60-965 Poznań

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu systemów mikroprocesorowych, programowania strukturalnego i obiektowego. Wymagana jest ponadto umiejętność korzystania ze środowiska projektowania dostarczanego przez producentów sprzętu elektronicznego takich jak Atmel (Atmel Studio) lub Xilinx (ISE, XPS, Vivado) oraz umiejętność pozyskiwania informacji ze wskazanych źródeł. Niezbędna jest również umiejętność poszerzania posiadanej wiedzy oraz pracy w zespole. Ze względu na kompetencje społeczne student powinien być świadomy, że wiedza w informatyce szybko staje się przestarzała i wymaga ustawicznego poszerzania. Student powinien prezentować postawę uczciwości, kreatywności, rzetelności, ciekawości poznawczej oraz okazywać szacunek dla innych ludzi.



Cel przedmiotu

1. Przekazanie studentom poszerzonej wiedzy z zakresu projektowania aplikacji wbudowanych na mikrokontrolerach jednoukładowych popularnych rodzin. Omówienie typowych urządzeń peryferyjnych spotykanych w mikrokontrolerach ze specyfiką ich pracy, konfigurowania i stosowania.
2. Wykształcenie umiejętności doboru optymalnej, dla danego zadania realizacji aplikacji z użyciem mikrokontrolera, z obsługą dedykowanych modułów peryferyjnych i przy uwzględnieniu wymogów związanych z oszczędnością energii i wydajnością obliczeniową.
3. Rozwijanie u studentów umiejętności rozwiązywania problemów doboru kontrolera i wdrożenia do realizacji typowych zadań z zakresu pobierania przetwarzania sygnałów, komunikacji z użytkownikiem bądź sterowania.
4. Kształtowanie u studentów umiejętności pracy zespołowej poprzez realizację elementów projektu i połączenie ich w całość.

Przedmiotowe efekty uczenia się

Wiedza

1. Absolwent zna szczegółowo budowę narzędzi wspomagających rozbudowę systemów wbudowanych. - [K2st_W1]
2. Absolwent posiada zaawansowaną wiedzę dotyczącą rozbudowy systemów wbudowanych. - [K2st_W3]
3. Absolwent zna szczegółowo wybrane standardy komunikacji pozwalające na rozbudowę systemów wbudowanych. - [K2st_W5]
4. Absolwent zna wiodące trendy w dziedzinie systemów wbudowanych odnośnie stosowanych układów scalonych wysokiej skali integracji. - [K2st_W6]

Umiejętności

1. Absolwent potrafi korzystać z systemów informatycznych wspomagających pracę w zespole i komunikację między użytkownikami. - [K2st_U2]
2. Absolwent potrafi wykorzystać wcześniej zdobyte umiejętności z projektowania elektroniki cyfrowej w celu realizacji określonego zadania związanego z rozbudową systemu wbudowanego. - [K2st_U5]
3. Absolwent potrafi dopasować nowoczesne narzędzia w celu realizacji określonego zadania związanego z rozbudową systemu wbudowanego. - [K2st_U6]
4. Absolwent zna ograniczenia wybranych narzędzi wspomagających rozbudowę systemów wbudowanych i potrafi ocenić ich przydatność. - [K2st_U9]
5. Absolwent potrafi rozwijać własne narzędzia na potrzeby realizacji projektu związanego z rozbudową systemu wbudowanego. - [K2st_U11]



Kompetencje społeczne

1. Absolwent jest gotów do ciągłego poszerzania wiedzy w obszarze systemów wbudowanych, zwłaszcza w zakresie zmieniających się trendów rozwoju rynku informatycznego. - [K2st_K1]
2. Absolwent rozumie konieczność stosowania najnowszych rozwiązań podczas implementacji mikrosystemów. - [K2st_K2]

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

a) w zakresie wykładów:

- w zakresie wykładów na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,

b) w zakresie laboratoriów:

- na podstawie oceny bieżącego postępu realizacji zadań

Ocena podsumowująca:

a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- weryfikowanie założonych efektów kształcenia realizowane jest przez ocenę wiedzy i umiejętności na podstawie kartkówek

b) w zakresie projektów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę sprawozdania przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole; dotyczy ćwiczeń projektowych o charakterze odtwórczym (student realizuje ćwiczenie według dostarczonej instrukcji)

- ocena realizacji złożonego zadania wymagającego integracji zdobytej w trakcie zajęć projektowych wiedzy i umiejętności; ocenie podlegają aspekty techniczne realizacji, umiejętność rozwiązania niekonwencjonalnych problemów oraz biegłość wykorzystania dostępnych narzędzi projektowych

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- omówienia dodatkowych aspektów zagadnienia,

- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,

- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,

- uwagi związane z udoskonaleniem materiałów dydaktycznych,



- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Pojęcia podstawowe w opisie architektury mikrokontrolerów: rejestr, przestrzeń adresowa, urządzenia wejścia/wyjścia, architektura harwardzka, cykl rozkazowy, DMA, przerwania, pamięć programu, sekcja bootloadera. Urządzenia kontrolnopomiarowe: mikrokontroler, mikroprocesor, ASIC, FPGA. Przegląd producentów i rodzin 8-bitowych oraz 32-bitowych kontrolerów. Mikrokontrolery 8-bitowe z rodziny AVR, obszary zastosowań, architektura, realizacja portów wejścia/wyjścia. Sposoby integrowania kontrolerów z urządzeniami wejścia/wyjścia (obsługa klawiatur, wyświetlaczy, układów regulowanych dwustawnie). Przetwarzanie analogowo cyfrowe z użyciem mikrokontrolerów. Przetwarzanie cyfrowo-analogowe z wykorzystaniem mikrokontrolerów. Niezawodność przetwarzania (układ Watchdog, układ Brown-out, Black out). Układy peryferyjne: timery liczniki, układy komunikacji z peryferiami I2C oraz SPI. Układy interfejsów komunikacyjnych: szeregowy, Modbus, CAN, Ethernet. Interfejs JTAG. Organizacja kodu aplikacji, bootloader i jego zastosowania. Kontrolery 32-bitowe na przykładzie STM32 z rdzeniem M3, M4 lub M7. Obsługa wyświetlacza, urządzeń peryferyjnych. Omówienie otwartych systemów operacyjnych na architekturze Cortex (eCos FreeRTOS).

Zajęcia laboratoryjne prowadzone są w formie piętnastu 2-godzinnych ćwiczeń, odbywających się w laboratorium, poprzedzonych 2-godzinną sesją instruktazową na początku semestru. Ćwiczenia realizowane są przez 2-osobowe zespoły studentów.

Laboratoria obejmują:

Projektowanie prostych aplikacji z użyciem kontrolerów 8-bitowych oraz 32-bitowych na dedykowanych płytach ćwiczebnych. Aplikacje ukazują przypadki użycia omawianych na wykładach układów peryferyjnych, urządzeń wejścia/wyjścia, modułów komunikacyjnych. Układy kontrolerów współpracują z typowymi podzespołami komunikacji człowiek-maszyna, jednostkami pomiarowymi (np. sensory warunków środowiskowych) oraz elementami wykonawczymi automatyki. Śledzenie wykonania aplikacji za pośrednictwem interfejsów typu JTAG. Narzędzia open-source w programowaniu systemów wbudowanych. Przygotowanie narzędzi (toolchain) Programowanie i analiza wykonania programu z wykorzystaniem dostępnych narzędzi. Implementowanie aplikacji typu bare-metal do realizacji zadanej funkcjonalności. Implementowanie aplikacji na wybrany system operacyjny (np. eCos lub freeRTOS).

Metody dydaktyczne

Wykład: prezentacja multimedialna, wykład tradycyjny.

Zajęcia laboratoryjne: wykonywanie instrukcji, zadania praktyczne, praca w zespole.

Literatura



Podstawowa

1. Mikrokontrolery ATmega w praktyce, R. Baranowski, BTC, Wa-wa, 2005
2. Mikrokontrolery STM32 w praktyce, K. Paprocki BTC, Wa-wa 2009
3. Noty katalogowe wskazanych modeli kontrolerów firm Atmel oraz ST Microelectronics

Uzupełniająca

1. Wybrane artykuły z bazy wiedzy www.atmel.com ukazujące przypadki aplikowania kontrolerów.
2. Wybrane artykuły z bazy wiedzy www.st.com ukazujące przypadki aplikowania kontrolerów.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
łączy nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do kolokwium) ¹	40	2

¹niepotrzebne skreślić lub dopisać inne czynności