



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Systemy wbudowane [N2EiT1>SW]

Przedmiot

Kierunek studiów

Elektronika i telekomunikacja

Rok/Semestr

2/4

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

niestacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

15

Laboratorium

15

Inne (np. online)

0

Ćwiczenia

15

Projekty/seminaria

0

Liczba punktów ECTS

2,00

Koordynatorzy

dr inż. Krzysztof Arnold

krzysztof.arnold@put.poznan.pl

Wykładowcy

dr inż. Krzysztof Arnold

krzysztof.arnold@put.poznan.pl

dr inż. Sławomir Michalak

slawomir.michalak@put.poznan.pl

Wymagania wstępne

Student posiada uporządkowaną i podbudowaną matematycznie wiedzę z podstaw teorii obwodów, niezbędną do zrozumienia, analizy i oceny działania obwodów elektrycznych. Ma wiadomości z techniki analogowej i techniki cyfrowej w zakresie podstawowym. Potrafi pozyskiwać informacje z literatury w języku polskim i angielskim. Rozumie konieczność poszerzania własnej wiedzy i jest odpowiedzialny. Zachowuje się aktywnie na zajęciach i systematycznie rozwiązuje napotkane problemy.

Cel przedmiotu

Przedstawienie kierunków i postępu integracji zasobów w strukturach mikroprocesorowych. Poznanie i zrozumienie organizacji systemów wbudowanych. Poznanie zasady działania, właściwości i perspektyw rozwojowych wbudowanych układów peryferyjnych. Opanowanie umiejętności programowania modułów systemowych, integrowanych w mikrokontrolerach oraz umiejętności wykorzystania warstwy sprzętowej i programowej mikrokontrolerów.

Przedmiotowe efekty uczenia się

Wiedza:

Ma uporządkowaną wiedzę w zakresie architektury modułów wbudowanych w mikrokontrolerach. Rozumie zasady działania i tryby pracy wbudowanych modułów systemowych. Ma podstawową wiedzę w zakresie programowania i wykorzystania zasobów współczesnych mikrokontrolerów. Posiada wiadomości o trendach rozwojowych dotyczących mikrokontrolerów.

Umiejętności:

Potrafi wykorzystywać dane źródłowe, integrować nowe informacje, analizować je, a także formułować i uzasadniać opinie. Posiada umiejętność tworzenia oprogramowania systemów wbudowanych z wykorzystaniem asemblera.

Kompetencje społeczne:

Potrafi pracować w zespole i kreatywnie włączać się do prac projektowych dotyczących systemów mikroprocesorowych. Dostrzega zmiany wynikające z postępu technologicznego i rozumie konieczność uaktualniania wiedzy i ciągłego podnoszenia kompetencji zawodowych. Ma poczucie odpowiedzialności za rozwijane projekty.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Egzamin pisemny i/lub ustny końcowy weryfikuje wiedzę i zrozumienie w zakresie treści wykładu. Zawiera pytania problemowe otwarte o zróżnicowanej punktacji. Ocena końcowa z egzaminu: poniżej 50% liczby punktów możliwych do uzyskania - 2,0; od 50% - 3,0; od 60% - 3,5; od 70% - 4,0; od 80% - 4,5; od 90% - 5,0.

Ocena końcowa z ćwiczeń audytoryjnych jest średnią arytmetyczną ocen za bieżącą aktywność (przygotowanie do zajęć, zachowanie, zaangażowanie, nabywanie i wykazywanie umiejętności).

Ocena końcowa z laboratorium jest średnią arytmetyczną ważoną ocen za realizację zadań podstawowych i dodatkowych (przygotowanie do kolejnych zadań, zachowanie, zaangażowanie, utrwalanie umiejętności) oraz ocen za sprawozdania indywidualne lub zespołowe, zamykające zadania.

Wagę określa się na zajęciach wprowadzających. Zadania dodatkowe weryfikują umiejętności przy ubieganiu się o zaliczenie laboratorium lub podwyższenie oceny. Mogą obejmować kolokwium pisemne lub ustne. Skala dla ocen końcowych (ćwiczenia, laboratorium): do 2,75 włącznie - 2,0; powyżej 2,75 - 3,0; powyżej 3,25 - 3,5; powyżej 3,75 - 4,0; powyżej 4,25 - 4,5; powyżej 4,75 - 5,0. Zaliczenie poprawkowe ćwiczeń audytoryjnych i/lub laboratorium obejmuje część praktyczną (laboratorium) i kolokwium pisemne lub ustne.

Treści programowe

Wykład: Układy scalone VLSI. Architektura mikrokontrolerów RISC z rdzeniem AVR. Podstawowe zadania modułów wbudowanych w strukturze mikrokontrolera. Restart zasobów mikrokontrolera. Środowisko programistyczne, programatory i systemy docelowe. Wbudowane pamięci Flash. Programowanie ISP. Język asemblera dla mikrokontrolerów AVR. Grupy rozkazów i ich działanie. Tworzenie oprogramowania. Podprogramy i stos. Komunikacja wbudowanych modułów z otoczeniem mikrokontrolera. Inicjalizacja i tryby pracy portów równoległych. System przerwań w mikrokontrolerach AVR. Interfejsy szeregowo wbudowane w struktury AVR.

Ćwiczenia i laboratorium: Wprowadzenie do środowiska programistycznego AVR Studio i podstawy programowania w języku asemblera. Stosowanie komend mikrokontrolerów AVR. Przesyłanie danych w systemie wbudowanym z rdzeniem AVR. Warunkowe i bezwarunkowe rozgałęzienia programu. Rozkazy i programy arytmetyczno-logiczne. Opóźnienia programowe. Wykorzystanie stosu.

Uruchamianie programu głównego i podprogramów z wykorzystaniem narzędzi symulacyjnych. Symulacja działania modułów systemu wbudowanego. Inicjalizacja modułów wbudowanych.

Wykorzystanie mikroprocesorowych modułów ewaluacyjnych. Uruchamianie portów równoległych w systemie docelowym. Obsługa wejść dwustanowych w trybie przeglądania. Zastosowania trybu wyjściowego portów do sterowania urządzeń zewnętrznych. Uruchamianie wybranych wbudowanych modułów transmisji szeregowo z wykorzystaniem sprzętu pomiarowego.

Metody dydaktyczne

Wykład z prezentacją multimedialną, wspomagany dyskusją problemową i przykładami na tablicy.

Ćwiczenia: wykonywanie zadań zleczonych przez prowadzącego, poprzedzonych wprowadzeniem,

z wykorzystaniem tablicy, środowiska programistycznego i narzędzi audiowizualnych.
Laboratorium: realizacja zadań problemowych podanych przez prowadzącego i weryfikacja wyników z wykorzystaniem środowiska programistycznego i zestawów uruchomieniowych, wprowadzanie zadań wymagających współpracy kilku zespołów.

Literatura

Podstawowa

1. Andrzej Pawluczuk: Sztuka programowania mikrokontrolerów AVR. Podstawy. Wyd. BTC, Warszawa 2006
2. Andrzej Pawluczuk: Sztuka programowania mikrokontrolerów AVR. Przykłady. Wyd. BTC, Warszawa 2007
3. Rafał Baranowski: Mikrokontrolery AVR ATmega w praktyce, Wyd. BCT, Warszawa 2005
4. ATmega16A. 8-bit AVR Microcontroller with 16K Bytes In-System Programmable Flash. Datasheet, Atmel Corporation 2014

Uzupełniająca

1. Paweł Hadam: Projektowanie systemów mikroprocesorowych, Wyd. BTC, Warszawa 2004
2. ATmega128A. 8-bit AVR Microcontroller Datasheet Complete. Atmel Corporation 2015
3. Jacek Bogusz: Lokalne interfejsy szeregowy w systemach cyfrowych. Wyd. BCT, Warszawa 2004

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	125	2,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	55	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	70	0,00