



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Aplikacje sprzętowe mikrokontrolerów i mikrokomputerów [S1EiT1>ASMiM]

Przedmiot

Kierunek studiów

Elektronika i telekomunikacja

Rok/Semestr

3/5

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

15

Laboratorium

30

Inne (np. online)

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

3,00

Koordynatorzy

dr inż. Krzysztof Arnold

krzysztof.arnold@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student posiada uporządkowaną i podbudowaną matematycznie wiedzę z podstaw teorii obwodów, niezbędną do zrozumienia działania obwodów elektrycznych. Opanował wiadomości o układach analogowych i cyfrowych w zakresie podstawowym. Ma wiedzę w o podstawach metrologii i sprzęcie pomiarowym. Potrafi pozyskiwać informacje z literatury w języku polskim i angielskim. Właściwie korzysta z aplikacji podstawowych elementów i układów elektronicznych. Potrafi przeprowadzić pomiary typowych parametrów sygnałów elektrycznych. Rozumie konieczność poszerzania własnej wiedzy i jest odpowiedzialny. Zachowuje się aktywnie na zajęciach, systematycznie rozwiązuje problemy w zespole.

Cel przedmiotu

Przedstawienie podstawowych zastosowań mikrokontrolerów i mikrokomputerów. Poznanie i zrozumienie mechanizmów współpracy mikrokontrolera i mikrokomputera z otoczeniem. Poznanie zasad działania mikrokontrolera w trybach przeglądania stanu modułów I/O i urządzeń zewnętrznych oraz obsługi przerwań. Opanowanie umiejętności programowania obsługi podstawowych układów, współpracujących z mikrokontrolerami. Opanowanie umiejętności tworzenia prostych systemów mikroprocesorowych.

Przedmiotowe efekty uczenia się

Wiedza:

Ma uporządkowaną wiedzę w zakresie architektury mikrokontrolerów i mikrokomputerów. Ma podstawową wiedzę w zakresie programowania mikrokontrolerów w języku asemblera, ich aplikacji sprzętowych oraz systemów mikroprocesorowych. Wie o trendach rozwojowych dotyczących współczesnych mikrokontrolerów i mikrokomputerów.

Umiejętności:

Potrafi korzystać z opisów firmowych współczesnych mikrokontrolerów. Umie posługiwać się środowiskiem programistycznym i pisać proste programy w języku asemblera. Potrafi analizować i zestawiać typowe systemy, budowane z wykorzystaniem mikrokontrolerów i mikrokomputerów.

Kompetencje społeczne:

Potrafi realizować projekty zespołowe. Ma poczucie odpowiedzialności za uruchamianie aplikacje mikrokontrolerów i mikrokomputerów.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Kolokwium pisemne i/lub ustne końcowe weryfikuje wiedzę i zrozumienie w zakresie treści wykładu. Zawiera pytania problemowe otwarte o zróżnicowanej punktacji. Ocena końcowa z kolokwium: 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 laboratorium jest średnią arytmetyczną ważoną ocen za realizację zadań podstawowych i dodatkowych (przygotowanie do kolejnych zadań, zachowanie, zaangażowanie, utrwalanie umiejętności) i 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: 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 laboratorium obejmuje część praktyczną i kolokwium pisemne lub ustne.

Treści programowe

Wykład: Architektura mikrokontrolerów. Zasada działania CPU. Środowisko programistyczne, programatory i systemy docelowe. Programowanie ISP. Programowanie mikrokontrolerów w języku asemblera. Organizacja stosu i wywoływanie podprogramów. Zasada działania, inicjalizacja i tryby pracy portów równoległych. System przerwań. Aplikacje i obsługa typowych układów wprowadzania danych do mikrokontrolera. Podstawowe aplikacje mikrokontrolerów i mikrokomputerów z obsługą urządzeń wyjściowych. Aplikacje z podglądem stanu portów równoległych i szeregowych. Komunikacja mikrokontrolerów i mikrokomputerów z terminalem znakowym i komputerem PC.

Laboratorium: Wprowadzenie do środowiska programistycznego i podstawy programowania w języku asemblera. Stosowanie komend mikrokontrolerów. Rejestry robocze, pamięci i przesyłanie danych. Warunkowe i bezwarunkowe zmiany kolejności wykonywania rozkazów. Rozkazy i programy arytmetyczno-logiczne. Opóźnienia programowe. Wykorzystanie stosu. Uruchamianie programu głównego i podprogramów z wykorzystaniem narzędzi symulacyjnych. Programowanie mikrokontrolera w systemie docelowym. Praca z zestawem ewaluacyjnym. Uruchamianie portów równoległych mikroprocesora i mikrokomputera. Obsługa wejść dwustanowych w trybie przeglądania. Obsługa przycisków i klawiatur. Zastosowania trybu wyjściowego portów do sterowania urządzeń zewnętrznych. Wizualizacja stanu portów równoległych z wykorzystaniem diod LED i wyświetlaczy. Uruchamianie wbudowanych modułów transmisji szeregowej z użyciem sprzętu pomiarowego. Uruchamianie komunikacji stacji podrzędnej z komputerem PC.

Tematyka zajęć

Wykład: Architektura mikrokontrolerów RISC z rdzeniem AVR. Zasada działania CPU. Środowisko programistyczne, programatory i systemy docelowe. Programowanie ISP. Programowanie mikrokontrolerów AVR w języku asemblera. Organizacja stosu i wywoływanie podprogramów. Zasada działania, inicjalizacja i tryby pracy portów równoległych. System przerwań w mikrokontrolerach AVR. Aplikacje i obsługa typowych układów wprowadzania danych do mikrokontrolera. Podstawowe aplikacje

mikrokontrolerów i mikrokomputerów z obsługą urządzeń wyjściowych. Aplikacje z podglądem stanu portów równoległych. Zasada działania i zastosowania wbudowanego modułu USART. Komunikacja mikrokontrolerów i mikrokomputerów z terminalem znakowym i komputerem PC.

Laboratorium: Wprowadzenie do środowiska programistycznego AVR Studio i podstawy programowania w języku asemblera. Stosowanie komend mikrokontrolerów AVR. Rejestry robocze, pamięci i przesyłanie danych. Warunkowe i bezwarunkowe zmiany kolejności wykonywania rozkazów. Rozkazy i programy arytmetyczno-logiczne. Opóźnienia programowe. Wykorzystanie stosu. Uruchamianie programu głównego i podprogramów z wykorzystaniem narzędzi symulacyjnych. Programowanie mikrokontrolera w systemie docelowym. Praca z zestawem ewaluacyjnym. Uruchamianie portów równoległych mikroprocesora i mikrokomputera. Obsługa wejść dwustanowych w trybie przeglądania. Obsługa przycisków i klawiatur. Zastosowania trybu wyjściowego portów do sterowania urządzeń zewnętrznych. Wizualizacja stanu portów równoległych z wykorzystaniem diod LED i wyświetlaczy. Uruchamianie wbudowanych modułów USART z użyciem sprzętu pomiarowego. Uruchamianie komunikacji stacji podrzędnej z komputerem PC.

Metody dydaktyczne

Wykład z prezentacją multimedialną, wspomagany dyskusją problemową i przykładami na tablicy, zawierający odniesienia do etapowych wyników zadań laboratoryjnych.

Laboratorium: realizacja praktycznych zadań problemowych podanych przez prowadzącego i weryfikacja wyników z wykorzystaniem środowiska programistycznego i zestawów uruchomieniowych, dyskusja porównawcza końcowych rozwiązań, ewentualne wprowadzanie zadań wymagających współpracy dwóch lub 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. BTC, 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. ATmega8A, mega AVR Data Sheet. 2020 Microchip Technology Inc.
4. Jacek Bogusz: Lokalne interfejsy szeregowy w systemach cyfrowych. Wyd. BTC, Warszawa 2004

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	90	3,00
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)	35	1,00