**Nazwa przedmiotu:**

Systemy wbudowane

**Koordynator przedmiotu:**

Henryk DOBROWOLSKI

**Status przedmiotu:**

Fakultatywny ograniczonego wyboru

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

SWB

**Semestr nominalny:**

7 / rok ak. 2015/2016

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

30 godz. wykładu
15 godz. zajęć w laboratorium
15 godz. projektu (konsuktacje i weryfikacja postępów)
Praca samodzielna - przygotowanie do laboratorium (20 godzin) oraz wykonanie projektu (40 godz.)
Łącznie 120 godz. - 4 ECTS

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

30 godz. wykładu
15 godz. laboratorium
15 godz. projektu
W sumie 60 godz. - 2 ECTS

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

1. Przygotowanie do laboratorium - w tym zapoznanie się z dokumentacją sprzętu, środowiskiem budowania oprogramowania oraz używanym systemem operacyjnym (MQX) - ok. 20 godzin
2. Praca nad projektem - w tym: zdefiniowanie założeń i wymagań, opis przewidywanego działania systemu (diagramy maszyn stanowych), kodowanie, uruchamianie i testy oraz sprawozdanie (dokumentacja projektu) - łącznie ok 40 godz.

W sumie 60 godz. - 2 ECTS

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład:  | 30h |
| Ćwiczenia:  | 0h |
| Laboratorium:  | 15h |
| Projekt:  | 15h |
| Lekcje komputerowe:  | 0h |

**Wymagania wstępne:**

- znajomość programowania strukturalnego z użyciem języka C
- znajomość podstaw budowy układów cyfrowych, w tym mikroprocesorów
- znajomość podstaw inżynierii oprogramowania

**Limit liczby studentów:**

28

**Cel przedmiotu:**

• zapoznanie ze specyfiką architektury i projektowania systemów wbudowanych, ze szczególnym naciskiem na systemy stosowane w (szeroko pojmowanym) przemyśle
• zapoznanie z podstawowymi standardami obowiązującymi dla tych systemów, również w aspekcie systemów klasy safety critical
• przedstawienie podstawowych wymagań związanych z projektowaniem oprogramowania dla systemów wbudowanych, z uwzględnieniem wymagań związanych z działaniem w czasie rzeczywistym

**Treści kształcenia:**

1. Podstawowe pojęcia
• System wbudowany: System wbudowany a system ogólnego zastosowania. Co jest a co nie jest systemem wbudowanym – "zamazana" granica.
• Reaktywność: Reaktywność a interaktywność. Środowisko i jego wymagania. Sprzężenie zwrotne – zmiany środowiska powodowane przez system reaktywny.
• Czas rzeczywisty: Definicja. Wymagania dla sprzętu i (zwłaszcza) oprogramowania. Rola systemu operacyjnego.
• Wzajemnie kolidujące wymagania: Koszt – wymagania środowiskowe – wydajność – równoważność sprzętu i oprogramowania. Niezawodność i wiarygodność. Dostępność (availability), MTBF, MTTR.
• Zastosowania / typy systemów: Systemy konsumpcyjne, przemysłowe (produkcyjne), pojazdów i statków powietrznych, militarne. Systemy krytyczne dla bezpieczeństwa i systemy krytyczne dla misji.
• Aspekty biznesowe: Podstawowe pojęcia: koszty i czas (NRE, TTM), projektowanie i mityczny osobomiesiąc

2. Prosty system mikroprocesorowy
• Mikrokontrolery: Cechy charakterystyczne. Mikrokontrolery uniwersalne i specyficzne aplikacyjnie. Integracja pamięci i modułów we-wy. Przykłady.
• Podstawowe interfejsy i moduły we/wy: Porty. Interfejsy szeregowe (w tym SPI, I2C) i równoległe. Przetworniki ADC/DAC.
• Systemy uruchomieniowe i narzędzia programowe: Host a system docelowy. Symulacja i jej ograniczenia. Kompilacja skrośna i problemy debugowania.
• Studium przypadku: Z8 Encore
Opis mikrokontrolera, system uruchomieniowy, własności środowiska przygotowania aplikacji.

3. Projektowanie
• Czas życia i jego modele
Model wodospadu i spirali. Model V (projektowanie). Model Harmony. Cykl życia systemu i procesy cyklu życia – odniesienia do standardów ISO/IEC 12207 i ISO/IEC 15288.
• Modele przetwarzania (MoC) w systemach wbudowanych
Pożądane własności. Procesy i sygnały. Modele bezczasowe, synchroniczne i z czasem dyskretnym.
• Formalizmy do modelowania w projektowaniu systemów wbudowanych
Maszyny stanowe (FSM, FSMD, i inne modele skończenie stanowe, w tym Statecharts Harela). Sieci Petriego. Języki modeli synchronicznych (przykład: Esterel)
• Zastosowanie UML i SysML
Diagramy i ich zastosowanie; aspekty czasu rzeczywistego. SysML – krótki tutorial. Wybrane studium przypadku. Modele wykonywalne i xUML.

4. Sprzęt
a) Współprojektowanie sprzętu i oprogramowania; SoC
• Równoważność HW/SW
• Technologia współprojektowania systemów, ewolucja współprojektowania HW/SW
• Rdzenie IP, komponenty wirtualne, platformy i programowalne platformy
• Zarys projektowania SoC
b) Technologie w systemach wbudowanych
• procesory
Architektury (Princeton/Harvard, ISA i CISC/RISC, wieloprocesory, specjalizacja – grafika, DSP, przykłady.
• pamięć: Rola ROM i RAM; implementacja (EPROM, EEPROM i Flash, SRAM a DRAM)
• magistrale i interfejsy (we-wy): wejście/wyjście: typy i adresowanie, przerwania, DMA; klasyfikacje magistral, studium przypadku: VME-32
• sensory i efektory: kategoryzacja i przykłady sensorów; proste przykłady efektorów: silnik krokowy, wyświetlacz LCD; sterowanie: regulatory z otwartą i zamkniętą pętlą, implementacja regulatora PID
c) Komunikacja systemów rozproszonych specyficzna dla systemów wbudowanych
• systemy "konsumenckie" – "inteligentny" dom, pervasive computing: Studium przypadku: LON
• systemy przemysłowe (produkcja) - magistrale terenowe (fieldbus): Studium przypadku: ASI, Profibus/ProfiNet
• systemy pojazdów i statków powietrznych (awionika): Studium przypadku: CAN i MIL-STD-1553B
• sieci sensoryczne: Wprowadzenie: zastosowania, architektury, problemy (z naciskiem na komunikację)

5. Oprogramowanie (systemowe)
• systemy operacyjne czasu rzeczywistego (RTOS): Wymagania dla obsługi przerwań i planisty. Jądro monolityczne czy mikrojądro – problem skalowalności. Mechanizmy komunikacji i synchronizacji procesów/wątków. POSIX. Pobieżny przegląd przykładów: OS-9, QNX, VxWorks, MQX.
• zarządzanie zadaniami: Typy zadań, przełączanie kontekstu, priorytety, wywłaszczanie, problem inwersji priorytetów. Studium przypadku: scheduler w OS-9
• przegląd strategii (algorytmów) planisty: Strategie sprawiedliwe i priorytetowe. Strategie priorytetowe zadań cyklicznych (w tym RMS)

**Metody oceny:**

Kolokwia wykładowe (2), oceny z ćwiczeń laboratoryjnych + ocena z projektu.
Laboratorium obejmuje ćwiczenia dotyczące programowania systemu opartego o Freescale TWR-K70F120M z użyciem środowiska CodeWarrior Development Studio (język C)
Projekt: Specyfikacja i wymagania dla podanego zadania; projekt architektury oprogramowania systemu w UML/SysML; implementacja wskazanych fragmentów (funkcjonalności) w C dla sprzętu laboratoryjnego TWR-K70F120M, z zastosowaniem s.o. MQX.

**Egzamin:**

nie

**Literatura:**

1. Frank Vahid, Tony Givargis: Embedded System Design: A Unified Hardware/Software Introduction. Wiley 2002
2. Richard Zurawski (ed): Embedded System Handbook, Taylor & Francis Group, 2006
3. Dokumentacja Freescale dla MCU Kinetis (K70), środowiska CodeWarrior, s.o. MQX i in. - materiały udostępniane studentom przez stronę przedmiotu
4. Inne: standardy i publikacje z IEEE, OMG, ACM itd.

**Witryna www przedmiotu:**

http://studia.elka.pw.edu.pl/

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt Wpisz opis:**

Zna podstawowe algorytmy planisty wykorzystywane w systemach wbudowanych czasu rzeczywistego

Weryfikacja:

Kolokwium

**Powiązane efekty kierunkowe:** K\_W10

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W04, T1A\_W05

### Profil ogólnoakademicki - umiejętności

**Efekt Wpisz opis:**

Student, który zaliczył przedmiot, potrafi: napisać oprogrogramowanie realizujące założoną funkcjonalność prostego systemu wbudowanego, również czasu rzeczywistego

Weryfikacja:

Projekt laboratoryjny

**Powiązane efekty kierunkowe:** K\_U15

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U15, T1A\_U16