



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Aplikacje internetu rzeczy [N1AiR2>PO8-AIR]

Przedmiot

Kierunek studiów

Automatyka i robotyka

Rok/Semestr

4/7

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

pierwszego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

niestacjonarne

Wymagalność

obieralny

Liczba godzin

Wykład

10

Laboratorium

20

Inne

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

3,00

Koordynatorzy

dr inż. Dominik Łuczak

dominik.luczak@put.poznan.pl

Wykładowcy

Wymagania wstępne

Wiedza: Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z elektroniki oraz podstaw programowania. Umiejętności: Student powinien posiadać umiejętność rozwiązywania podstawowych problemów z zakresu cyfrowego przetwarzania sygnałów oraz umiejętność pozyskiwania informacji ze wskazanych źródeł. Powinien również rozumieć konieczność poszerzania swoich kompetencji i być gotowy do podjęcia współpracy w zespole. Kompetencje społeczne: Ponadto w zakresie kompetencji społecznych student musi przejawiać takie cechy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawczą, kreatywność, kulturę osobistą, szacunek dla innych ludzi.

Cel przedmiotu

Celem przedmiotu jest poznanie podstaw teoretycznych, zasady działania i możliwych zastosowań urządzeń komunikujących się z innymi obiektami rozproszonego systemu sterowania bez udziału człowieka (ang. Machine to Machine, M2M) zarówno przewodowo jak i bezprzewodowo poprzez globalną sieć internetową lub sieć lokalną. Student po zakończeniu kształcenia powinien potrafić: 1) opracować prostą aplikację dla urządzenia brzegowego pozwalającą na zdalne zarządzanie wybranym przedmiotem podłączonym do globalnej sieci, 2) opracować urządzenie umożliwiające komunikację z innymi obiektami w sieci Internet lub sieci lokalnej, 3) przygotować interfejs użytkownika do zdalnego zarządzania jednym i kilkoma obiektami.

Przedmiotowe efekty uczenia się

Wiedza:

1. Student ma uporządkowaną w zaawansowanym stopniu wiedzę w zakresie wybranych algorytmów i struktur danych oraz metodyki i technik programowania proceduralnego i obiektowego; - [K1_W8],
2. ma uporządkowaną wiedzę w zakresie architektur komputerów, systemów i sieci komputerowych oraz systemów operacyjnych w tym systemów operacyjnych czasu rzeczywistego; - [K1_W9],
3. zna i rozumie typowe technologie inżynierskie, zasady oraz techniki konstruowania prostych systemów automatyki i robotyki; zna i rozumie zasady doboru układów wykonawczych, jednostek obliczeniowych oraz elementów i urządzeń pomiarowo-kontrolnych - [K1_W20],
4. orientuje się w aktualnym stanie oraz najnowszych trendach rozwojowych obszaru automatyki i robotyki - [K1_W21],
5. zna i rozumie fundamentalne dylematy współczesnej cywilizacji powiązane z rozwojem automatyki i robotyki - [K1_W28].

Umiejętności:

1. Student potrafi zaprojektować i praktycznie wykorzystać proste układy diagnostyczno-decyzyjne dedykowane systemom automatyki i robotyki; - [K1_U21],
2. potrafi dobrać rodzaj i parametry układu pomiarowego, jednostki sterującej oraz modułów peryferyjnych i komunikacyjnych dla wybranego zastosowania oraz dokonać ich integracji w postaci wynikowego systemu pomiarowo-sterującego - [K1_U22],
3. potrafi dokonać identyfikacji i sformułować specyfikację prostych zadań inżynierskich z zakresu automatyki i robotyki; - [K1_U23],
4. potrafi opracować rozwiązanie prostego zadania inżynierskiego oraz zaimplementować, przetestować i uruchomić go w wybranym środowisku programistycznym na komputerze klasy PC dla wybranych systemów operacyjnych - [K1_U26].

Kompetencje społeczne:

1. Student posiada świadomość ważności i rozumie pozatechniczne aspekty i skutki działalności inżynierskiej, w tym jej wpływ na środowisko i związaną z tym odpowiedzialność za podejmowane decyzje; jest gotów do dbałości o dorobek i tradycje zawodu; - [K1_K2]
2. posiada świadomość konieczności profesjonalnego podejścia do zagadnień technicznych, skrupulatnego zapoznania się z dokumentacją oraz warunkami środowiskowymi, w których urządzenia i ich elementy mogą funkcjonować - [K1_K5]

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

a) w zakresie wykładów:

na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,

b) w zakresie laboratorium:

na podstawie oceny znajomości i zrozumienia bieżących zagadnień prezentowanych w ramach przedmiotu.

Ocena podsumowująca:

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

i. ocenę wiedzy i umiejętności wykazanych na teście wielokrotnego wyboru.

ii. omówienie wyników,

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

i. opracowanie systemu kontrolno-pomiarowego IoT w wersji podstawowej.

ii. opracowanie systemu kontrolno-pomiarowego IoT w wersji rozszerzonej.

Uzyskiwanie dodatkowych punktów za aktywność podczas zajęć, w szczególności za:

- i. samodzielną budowę modułu elektronicznego systemu kontrolno-pomiarowego i opracowanie dokumentacji
- ii. efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanych problemu
- iii. uwagi związane z udoskonaleniem materiałów dydaktycznych.

Treści programowe

Przedmiot obejmuje szeroki zakres zagadnień związanych z Internetem Rzeczy (IoT), od podstawowych elementów i zadań aplikacji, po budowę systemów, moduły rozwojowe, środowiska programistyczne i formaty wymiany danych. Studenci poznają również aplikacje webowe z interfejsem RESTful, konfigurację systemów operacyjnych Linux z wybranymi elementami sprzętowymi np. GPIO, SPI, UART, I2C, tworzenie aplikacji serwerowej na urządzenia o ograniczonych zasobach, strukturę i interakcję interfejsów webowych (np. HTML, jQuery, CSS, Bootstrap), wybrane protokoły sieciowe (np. Ethernet, IP, TCP, UDP, DHCP, ARP, HTTP, MQTT) oraz zagadnienia bezpieczeństwa.

Zajęcia laboratoryjne dają studentom możliwość praktycznego zastosowania zdobytej wiedzy. Obejmują one zapoznanie się z aparaturą, zasadami BHP, środowiskiem projektowym, formatem JSON, wykonanie interfejsu RESTful, konfigurację systemu Linux, opracowanie aplikacji na system Linux do obsługi wybranych interfejsów GPIO, SPI, UART, I2C, projekt interfejsu użytkownika, wykonanie interfejsów webowych (np. HTML, jQuery, Bootstrap) z elementami interaktywnymi, z możliwością zastosowania algorytmów przetwarzania sygnałów oraz opracowanie i testy układu kontrolno-pomiarowego.

Program kursu zapewnia kompleksowe przygotowanie do pracy z aplikacjami Internetu Rzeczy. Studenci zdobędą wiedzę teoretyczną i praktyczne umiejętności niezbędne do projektowania, tworzenia i implementacji urządzeń brzegowych IoT w różnych dziedzinach.

Tematyka zajęć

Program wykładu obejmuje następujące zagadnienia:

1. Elementy i zadania aplikacji Internetu rzeczy. Format wymiany danych między systemami - JSON, interfejs REST (representational state transfer), klient HTTP.
2. Aplikacja REST w PHP. Serwer HTTP na urządzeniu IoT i REST.
3. Konfiguracja systemu operacyjnego Linux - konfiguracja GPIO, SPI, UART, I2C.
4. System Linux - prawa dostępu, skrypty shell, tworzenie użytkowników, dostęp zdalny SSH.
5. Struktura interfejsu webowego – HTML, jQuery. Interakcja interfejsu webowego - JS, jQuery
6. Prezentacja interfejsu webowego - CSS, Bootstrap.
7. Wybrane protokoły komunikacyjne (np. Ethernet, IP, TCP, UDP, TLS, DHCP, ARP, HTTP, MQTT).

Program zajęć laboratoryjnych obejmuje następujące zagadnienia:

1. Zajęcia organizacyjne - zapoznanie z aparaturą i przypisami BHP, wstęp do środowiska projektowego. Format wymiany danych JSON.
2. Parsowanie danych JSON.
3. Aplikacja REST w PHP. Serwer HTTP na urządzeniu brzegowym IoT.
4. Klient HTTP.
5. Interfejs wiersza poleceń CLI (Command Line Interface) dla macierzy diod LED. REST interfejs dla macierzy diod LED.
6. Interfejs wiersza poleceń dla sensorów. REST interfejs dla sensorów.
7. Interfejs I2C. REST interfejs dla sensora z interfejsem I2C.
8. Interfejs WWW użytkownika - dynamiczna generacja struktury interfejsu dla sensorów (jQuery i HTML).
9. Interfejs WWW użytkownika - dynamiczna generacja struktury interfejsu dla macierzy diod LED (jQuery i HTML).
10. Interfejs WWW użytkownika - warstwa prezentacji (CSS i Bootstrap).
11. Aplikacja końcowa obejmująca podstawową aplikację z możliwością rozszerzenia o dodatkowe moduły np. MQTT, WS, HTTPS, CRON, komputer jednopłytkowy poprzez SPI lub UART zarządza płytą NUCLEO-ST32F7 realizującą aplikację automatycznej regulacji.

Metody dydaktyczne

1. Wykład: prezentacja multimedialna ilustrowana symulacjami komputerowymi i rzeczywistym

systemem

2. Zajęcia laboratoryjne: wykorzystanie jednopłytkowych modułów uruchomieniowych z zestawem czujników, środowiska programistyczne dla aplikacji Internetowych

Literatura

Podstawowa

1. Ilya Grigorik, Wydajne aplikacje internetowe. Przewodnik, Helion, 2014 / Ilya Grigorik, High Performance Browser Networking, 2013
2. Justin Hutchens, Skanowanie sieci z Kali Linux : receptury : bezpieczeństwo sieci w Twoich rękach!, Helion, 2015 / Justin Hutchens, Kali Linux Network Scanning Cookbook, 2014
3. Adam Gerber, Clifton Craig, Android Studio : wygodne i efektywne tworzenie aplikacji, Helion, 2016 / Adam Gerber, Clifton Craig, Learn Android Studio Build Android Apps Quickly and Effectively, 2015
4. Adrian McEwen, Hakim Cassimally, Designing the Internet of Things, Wiley, 2013.

Uzupełniająca

1. Łuczak, D. Machine Fault Diagnosis through Vibration Analysis: Continuous Wavelet Transform with Complex Morlet Wavelet and Time–Frequency RGB Image Recognition via Convolutional Neural Network. Electronics 2024, 13, 452, doi:10.3390/electronics13020452.
2. Łuczak, D.; Brock, S.; Siembab, K. Cloud Based Fault Diagnosis by Convolutional Neural Network as Time–Frequency RGB Image Recognition of Industrial Machine Vibration with Internet of Things Connectivity. Sensors 2023, 23, 3755, doi:10.3390/s23073755.
3. Łuczak D., Remote laboratory with WEB interface, Computer Applications in Electrical Engineering, Vol. 9, str. 257-268, Poznań, 2011, ISSN 1508-4248.
4. Karol Rogowski, Świat poza jQuery : biblioteki : AngularJS, KnockoutJS i BackbonesJS, Wydawnictwo Naukowe PWN, 2014.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	75	3,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	1,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	45	2,00