

Przykładowe tematy zadań inżynierskich dla przedmiotu Systemy wbudowane

Edycja 2021/2022, wersja 220228.0

Zadania inżynierskie dotyczą praktycznej realizacji urządzeń wbudowanych. Działanie wykonanego urządzenia należy przedstawić osobie prowadzącej zajęcia, opracować dokumentację z realizacji zadania (sprawozdanie) oraz zaprezentować działanie wykonanego urządzenia. Preferowanym sposobem prezentacji jest kilkuminutowy film wideo (najlepiej z napisami) na temat opracowanego systemu wbudowanego.

Przypominam składniki oceny wg sylabusu przedmiotu:

Ocena z zajęć jest wyliczana na podstawie czterech niżej wymienionych składników. Każdy składnik ma swoją wagę, która jest uwzględniana przy wyliczaniu oceny końcowej.

1. *Praca nad powierzonym zadaniem. Waga składnika: 20%.*
 - a. *W przypadku pracy w laboratorium: samodzielne przygotowanie do pracy w laboratorium, profesjonalizm i systematyczność w podejściu do realizacji powierzonego zadania.*
 - b. *W przypadku pracy samodzielnej: profesjonalizm i systematyczność w podejściu do realizacji powierzonego zadania i raportowaniu postępów.*
2. *Poziom merytoryczny i stopień realizacji założonej funkcjonalności praktycznego zadania inżynierskiego w formie prototypu systemu wbudowanego. Waga składnika: 50%.*
3. *Poziom merytoryczny, poprawność i ogólna jakość (w tym estetyka) opracowanego sprawozdania z prac przeprowadzonych w ciągu semestru. Waga składnika: 15%.*
4. *Poziom merytoryczny i jakość prezentacji opracowanego rozwiązania. Waga składnika: 15%.*

Zalecam, aby na przedostatnie spotkanie mieć gotowy zarówno działający model systemu oraz sprawozdanie. Natomiast na ostatnim spotkaniu będziemy mieć możliwość zaprezentowania naszych produktów pozostałym osobom z grupy - spotkanie odbędzie się w formie mini-konferencji.

1 Dostępne do wykorzystania platformy sprzętowe:

(uwaga: liczba szt. ograniczona)

X-NUCLEO-IKS01A3 (dostępne 3 szt. u każdej z osób prowadzących)

<https://kamami.pl/stm-nucleo-shield/575087-x-nucleo-iks01a3-shield-z-czujnikami-ruchusrodowiskowymi-dla-stm32-nucleo.html>

STEVAL-MKSBOX1V1 (dostępne 2 szt. u każdej z osób prowadzących)

<https://www.st.com/en/evaluation-tools/steval-mksbox1v1.html>

X-NUCLEO-6180A1 (dostępne 2 szt. u każdej z osób prowadzących)

<https://kamami.pl/stm-nucleo-shield/579207-x-nucleo-6180a1-plytka-rozszerzen-z-czujnikiem-odleglosci-tof-vl6180.html>

X-NUCLEO-NFC04A1 (dostępne po 1 szt. u każdej z osób prowadzących)

<https://kamami.pl/stm-nucleo-shield/566120-x-nucleo-nfc04a1-plytka-rozszerzajaca-z-tagiem-nfc-rfid.html>

X-NUCLEO-NFC05A1 (dostępne po 1 szt. u każdej z osób prowadzących)

<https://kamami.pl/stm-nucleo-shield/566192-x-nucleo-nfc05a1-plytka-rozszerzeniowa-z-czytnikiem-nfc.html>

2 Przykłady ekonomicznych platform bazowych do samodzielnych eksperymentów:

- ESP32-DEVKITC-32D V4. Ciekawsze zastosowania: komunikacja sieciowa Wi-Fi i Bluetooth
- STM32F411E-DISCO. Ciekawsze zastosowania to odtwarzacze audio: Wave i MP3.
- NUCLEO-F429ZI. Ciekawsze zastosowania to urządzenia komunikujące się z Internetem przez interfejs Ethernet.
- Raspberry Pi Zero W lub Zero W 2.
- Raspberry Pi Pico.
- ESP32-LyraT. Platforma o dość dużych możliwościach, przydatna do zbudowania urządzeń embedded przetwarzających dźwięk, link:
<https://www.espressif.com/en/products/devkits/esp32-lyrat>

3 Przykłady ekonomicznych sensorów i układów wykonawczych:

- Akcelerometry, żyroskopy, a bardziej ogólnie jednostki *inertial measurement unit* (IMU)
- Sensor DHT11 lub DHT22 – pomiar temperatury i wilgotności względnej.
- Taśmy LED ze sterownikiem WS2812B.
- Serwomechanizm – miniaturowy, o niskim momencie obrotowym.

Przykłady zadań do realizacji

1 Programator STM32 na Raspberry Pi

Wymagania technologiczne:

- platforma sprzętowa programatora: Raspberry Pi.
- docelowa platforma sprzętowa: STM32 z rdzeniami Cortex-M

Oceniane aspekty:

- walory użytkowe,
- zaimplementowana przynajmniej część funkcjonalności,
- propozycja innowacyjności (potencjalnej lub zaimplementowanej),
- styl implementacji (przejrzystość kodu źródłowego, zastosowanie nowoczesnych bibliotek).

Oczekiwana funkcjonalność:

- Użyteczny i łatwy w obsłudze interfejs zewnętrzny,
- Zdalne zarządzanie konfiguracjami (własna aplikacja lub LwM2M)
- Szybkie przeprogramowywanie z wybranej, lokalnej kopii firmware'u.

Materiały pomocnicze:

- <https://learn.adafruit.com/programming-microcontrollers-using-openocd-on-raspberry-pi/overview>

2 Demonstracja działania klienta LwM2M na urządzeniu wbudowanym bez aplikacyjnego systemu operacyjnego.

Wymagania technologiczne:

- platforma sprzętowa: STM32, ESP32, ESP9266 (mile widziana implementacja dla >1 platformy, zależnie od możliwości czasowych i logistycznych),
- interfejs komunikacyjny – taki jak dostępny na platformie sprzętowej - do uzgodnienia z osobą prowadzącą,
- zastosowanie popularnych i aktywnie rozwijanych bibliotek (np. LwIP dla STM32)
- darmowe narzędzia do rozwoju oprogramowania, np. STM32CubeIDE, Arduino.

Oceniane aspekty:

- przekonująca i efektowna demonstracja działania,
- demonstracja możliwości zmiany stanu i odczytu modułów czujników / układów wykonawczych.

Oczekiwana funkcjonalność:

- interakcja z modułami wykonawczymi lub czujnikami – w miarę możliwości technicznych i logistycznych; jeśli czujniki nie będą dostępne, można rozważyć programową emulację obecności czujników i układów wykonawczych

3 Zdalna aktualizacja firmware'u na STM32

Wymagania technologiczne:

- platforma sprzętowa: STM32 dostępna w laboratorium (STM32F4, STM32F7),
- interfejs komunikacyjny: Ethernet,
- zastosowanie lwIP,
- język programowania: C,
- narzędzia GCC i Makefile lub STM32CubeIDE

Oceniane aspekty:

- przekonująca i efektowna demonstracja działania,
- walory użytkowe: czas wgrywania oprogramowania, przewidywalność,
- styl implementacji (przejrzystość kodu źródłowego).

Oczekiwana funkcjonalność:

- Przez aplikację z GUI lub stronę WWW łączymy się do urządzenia, w którym chcemy uaktualnić oprogramowanie i wysyłamy nowy plik z firmware'm.

Materiały:

https://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-expansion-packages/x-cube-sbsfu.html

4 Odtwarzacz plików dźwiękowych FLAC (Free Losless Audio Codec)

Wymagania technologiczne:

- platforma sprzętowa: płytk testowa STM32 z wyjściem audio,
- język programowania: C,
- darmowe narzędzia, preferowany STM32CubeIDE.

Oceniane aspekty:

- przekonująca i efektowna demonstracja działania,
- implementacja dekodowania FLAC,
- jakość dźwięku: brak zakłóceń i trzasków wynikających z nieprawidłowego dekodowania, płynność odtwarzania,
- styl implementacji (przejrzystość kodu źródłowego),
- walory użytkowe (interfejs użytkownika, czas reakcji).

Oczekiwana funkcjonalność: należy sprawdzić możliwość i zaimplementować w praktyce

odtworzenie dźwięku w formacie FLAC. Można bazować na wstępnie przygotowanym projekcie odtwarzającym dźwięk w formacie Wave. Interfejs użytkownika może być minimalny (play/pause/stop).

5 Demonstracja działania jednostki MMU w mikroprocesorze ARM

Wymagania technologiczne:

- platforma sprzętowa: mikroprocesor z aplikacyjnym rdzeniem ARM Cortex-A na płytce popularnej platformie sprzętowej (np. Raspberry Pi),
- język programowania: C,
- środowisko: darmowe; dla Raspberry Pi: GCC+Makefile, dla ZYBO: Xilinx Vivado.

Oceniane aspekty:

- przekonująca demonstracja działania,
- stopień uzyskania zadanej funkcjonalności,
- styl implementacji (przejrzystość kodu źródłowego).

Oczekiwana funkcjonalność: system demonstrujący działanie jednostki MMU w nowoczesnym mikroprocesorze ARM Cortex-A.

6 Odtwarzacz internetowych stacji radiowych

Wymagania technologiczne:

- platforma sprzętowa: płytka testowa ESP32,
- język programowania: C,
- darmowe (najlepiej na wolnej licencji) narzędzia do rozwoju oprogramowania.

Oceniane aspekty:

- przekonująca i efektowna demonstracja działania,
- styl implementacji (przejrzystość kodu źródłowego),
- walory użytkowe (interfejs użytkownika, jakość odtwarzanego dźwięku).

Oczekiwana funkcjonalność:

- odtwarzanie strumienia dźwiękowego pobieranego z przykładowej stacji radiowej,
- możliwość wprowadzania przez użytkownika adresów stacji,
- zapamiętywanie ostatnio używanej stacji oraz listy ulubionych stacji,
- regulacja głośności odtwarzanego dźwięku.

Zadania o charakterze inżynierskim

W tej klasie zadań posługujemy się nowoczesnymi, ale już popularnymi technologiami. Implementujemy przy ich pomocy taką funkcjonalność, która w wybranych aspektach prezentuje możliwości tych technologii. Tematy mają więcej aspektów inżynierskich niż badawczych. Przy ocenianiu kładziony jest nacisk na walory dydaktyczne realizacji zadania, czyli przede wszystkim:

- przejrzystość kodu i stosowanie dobrych praktyk przy pisaniu oprogramowania,
- przejrzysta dokumentacja o wysokich walorach informacyjnych, nastawiona na przekazywanie wiedzy czytelnikom również nie będącym specjalistami w dziedzinie,
- efektowna prezentacja działania.

Celem realizacji zadań jest osiągnięcie efektów kształcenia wyszczególnionych w sylabusie modułu.

1 Dydaktyczny proof-of-concept odtwarzacza plików dźwiękowych MP3

Wymagania technologiczne:

- platforma sprzętowa: płytka testowa STM32 z wyjściem audio,

- język programowania: C,
- darmowe (najlepiej na wolnej licencji) narzędzia do rozwoju oprogramowania.

Oceniane aspekty:

- przekonująca i efektowna demonstracja działania,
- styl implementacji (przejrzystość kodu źródłowego),
- jakość dźwięku: brak zakłóceń i trzasków wynikających z nieprawidłowego dekodowania, płynność odtwarzania,
- poprawne dekodowanie MP3,
- walory użytkowe (interfejs użytkownika).

Oczekiwana funkcjonalność: należy sprawdzić możliwość i zaimplementować w praktyce odtwarzanie dźwięku w formacie MP3 bazując na wstępnie przygotowanym projekcie odtwarzającym dźwięk w formacie Wave.

2 Kontroler MIDI

Wymagania technologiczne:

- platforma sprzętowa: płytk testowa z mikrokontrolerem STM32 i z dostępnym interfejsem USB zdolnym do pracy jako urządzenie USB,
- urządzenie powinno po dołączeniu do komputera zgłaszać się jako kontroler MIDI,
- język programowania: C,
- STM32CubeMX, STM32CubeIDE,

Oceniane aspekty:

- przekonująca i efektowna demonstracja działania – współpraca z aplikacją np. wirtualnego syntezatora,
- styl implementacji (przejrzystość kodu źródłowego),
- kreatywność przy wymyślaniu interfejsu użytkownika: minimum to klawiatura zbudowana z kilku przycisków, jednak urządzenie można rozbudować o czujniki położenia, czujniki odległości, etc., może ja także zbudować

Oczekiwana funkcjonalność: należy skonstruować urządzenie, które po dołączeniu do komputera zgłosi się jako kontroler MIDI. Interfejs użytkownika powinien pozwalać na generowanie przynajmniej kilku dźwięków.

Źródło inspiracji: <https://samjkent.co.uk/stm32f4-midi-brain/>

3 Eksperymentalny instrument muzyczny

Wymagania technologiczne:

- platforma sprzętowa: płytk testowa STM32 z wyjściem audio,
- język programowania: C,
- darmowe (najlepiej na wolnej licencji) narzędzia do rozwoju oprogramowania.

Oceniane aspekty:

- przekonująca i efektowna demonstracja działania,
- styl implementacji (przejrzystość kodu źródłowego),
- kreatywność przy wymyślaniu interfejsu użytkownika,
- walory użytkowe, np. subiektywny czas reakcji na wciśnięcie klawisza, polifonia.

Oczekiwana funkcjonalność: należy skonstruować urządzenie, które po dołączeniu słuchawek lub wzmacniacza audio będzie działało jak elektroniczny instrument muzyczny (prosty syntezator lub keyboard z ciekawym interfejsem użytkownika). Alternatywnie urządzenie może współpracować z kontrolerem MIDI przez interfejs USB.

4 Inteligentna kamera do monitoringu lub fotopułapka na Raspberry Pi wg ustalonej funkcjonalności

Wymagania technologiczne:

- platforma sprzętowa: Raspberry Pi.

Oceniane aspekty:

- walory użytkowe,
- zaimplementowana część funkcjonalności,
- propozycja innowacyjności (potencjalnej lub zaimplementowanej),
- styl implementacji (przejrzystość kodu źródłowego, zastosowanie nowoczesnych bibliotek).

Oczekiwana funkcjonalność:

- Użyteczny i łatwy w obsłudze interfejs zewnętrzny,
- Detekcja ruchu
- Integracja z zadaniem relatywnie prostym interfejsem bazującym na porcie szeregowym (odczyt poleceń, wysyłanie statusów)
- Prosta konfiguracja, np. tekst, XML
- Inspiracja oraz przykład narzędzia, które można zastosować: **motion** - <https://motion-project.github.io/index.html>

5 Cyfrowe przetwarzanie dźwięku – demonstrator

Wymagania technologiczne:

- platforma sprzętowa: jedna z platform dostępnych w laboratorium i wyposażonych w odpowiednie interfejsy audio,
- darmowe (najlepiej na wolnej licencji) narzędzia do rozwoju oprogramowania.

Oceniane aspekty:

- jakość dźwięku – czy nie ma szumów, trzasków, nieprzewidzianych zniekształceń?
- stopień realizacji zadania,
- styl implementacji (przejrzystość kodu źródłowego).

Oczekiwana funkcjonalność: przykłady, do wyboru:

- efekt gitarowy,
- procesor surround,
- efekty estradowe (echo, jeszcze lepiej reverb),
- wizualizacja dźwięku w czasie rzeczywistym.

6 Przykładowa sieć sensorowa z MySensors

Wymagania technologiczne:

- zastosowanie systemu MySensors.

Oceniane aspekty:

- przekonująca demonstracja działania,
- walory użytkowe,
- zawarte w sprawozdaniu kreatywne pomysły zastosowań,
- styl implementacji (przejrzystość kodu źródłowego).

Oczekiwana funkcjonalność: model sieci automatyki inteligentnego budynku z wybranymi czujnikami oraz modułami wykonawczymi.

Informacje dodatkowe: <https://www.mysensors.org/>

7 Energooszczędny węzeł sieci zgodny z systemem MySensors

Wymagania technologiczne:

- Platforma sprzętowa: STM32 nucleo-32,
- zastosowanie modułu radiowego nRF24L01,
- zasilanie z pakietu baterii 2xAA ,2xAAA lub 1x ogniwo Li-Ion 18650

Oceniane aspekty:

- przekonująca demonstracja działania,

- średni pobór energii,
- walory użytkowe (subiektywnie oceniany czas reakcji przy interakcji z użytkownikiem),
- styl implementacji (przejrzystość kodu źródłowego).

Oczekiwana funkcjonalność:

- Funkcjonalność węzła sieci MySensors,
- implementacja algorytmów zarządzania energią dla zapewnienia oszczędzania baterii/akumulatora,
- implementacja przykładowej funkcjonalności ustalonej z osobą prowadzącą zajęcia: modułu wykonawczego, modułu interakcji z użytkownikiem i/lub modułu sensorycznego.

8 Obsługa modułu RFID RC522

Wymagania technologiczne:

- platforma sprzętowa: STM32 dostępny w laboratorium (STM32F4, STM32F7),
- język programowania: C,
- darmowe (najlepiej na wolnej licencji) narzędzia do rozwoju oprogramowania.

Oceniane aspekty:

- przekonująca demonstracja działania,
- zawarte w sprawozdaniu kreatywne pomysły zastosowań,
- stopień uzyskania zadanej funkcjonalności,
- styl implementacji (przejrzystość kodu źródłowego).

Oczekiwana funkcjonalność: urządzenie demonstrujące działanie modułu RFID RC522 z mikrokontrolerem STM32 w przykładowej, użytecznej aplikacji, np. kontroli dostępu.

9 Serwer FTP z zabezpieczeniem “tamper”

Wymagania technologiczne:

- platforma sprzętowa: STM32 lub ESP32
- interfejs komunikacyjny: zależnie od dostępności na wybranej platformie
- język programowania: C/C++
- darmowe (najlepiej na wolnej licencji) narzędzia do rozwoju oprogramowania.

Oceniane aspekty:

- przekonująca demonstracja działania,
- stopień uzyskania zadanej funkcjonalności,
- walory użytkowe (np. możliwość połączenia się przez FileZilla i upload/download plików),
- implementacja dodatkowej funkcjonalności („kreatywny dodatek”),
- styl implementacji (przejrzystość kodu źródłowego).

Oczekiwana funkcjonalność:

- urządzenie wbudowane działające jako serwer FTP przechowujący pliki na dołączonej pamięci masowej: z interfejsem USB (pendrive) lub na karcie pamięci SD,
- Tamper: wykrywanie próby fizycznej ingerencji w serwer – kreatywne zastosowanie czujników ;)

10 Implementacja demo systemu z brokerem i klientami MQTT

Wymagania technologiczne:

- klient: platforma sprzętowa STM32 dostępna w laboratorium (STM32F4, STM32F7), ESP32, ESP8266.
- interfejs komunikacyjny - taki jak jest dostępny na zastosowanej platformie, np. Ethernet lub Wi-Fi,
- język programowania: C,
- darmowe (najlepiej na wolnej licencji) narzędzia do rozwoju oprogramowania.

Oceniane aspekty:

- przekonująca demonstracja działania, najlepiej dla dwóch klientów i przesyłaniem informacji zarówno od klienta jak i do klienta,
- stopień uzyskania zadanej funkcjonalności,
- zawarte w sprawozdaniu kreatywne pomysły zastosowań.

Oczekiwana funkcjonalność:

- system urządzeń komunikujących się przez protokół MQTT.

11 Proste lub zaawansowane przetwarzanie obrazu w czasie rzeczywistym

Wymagania technologiczne:

- platforma sprzętowa Raspberry Pi z dołączonym modułem kamery,
- biblioteka OpenCV.

Oceniane aspekty:

- styl implementacji – przejrzystość kodu źródłowego (bardzo ważne),
- przekonująca demonstracja działania,
- stopień uzyskania zadanej funkcjonalności,
- zawarte w sprawozdaniu kreatywne pomysły zastosowań.

Oczekiwana funkcjonalność:

- ciekawa demonstracja na Raspberry Pi z dołączonym modułem kamery, podgląd na monitorze;
- przykłady: detektor krawędzi, detektor ruchu, śledzenie poruszających się obiektów, klasyfikacja obiektów.

12 Interaktywna aplikacja demonstracyjna dla pojazdu opartego na podwoziu Magician Chassis - możliwość realizacji w miarę dostępności elementów mechanicznych

Wymagania technologiczne:

- platforma sprzętowa: STM32 Nucleo64 lub Arduino UNO R3, Arduino Motor shield, ultradźwiękowy czujnik odległości, czujnik odbiciowy TCRT5000, diody WS2812
- język programowania: C, Arduino,
- darmowe (najlepiej na wolnej licencji) narzędzia do rozwoju oprogramowania.

Oceniane aspekty:

- przekonująca i efektowna demonstracja działania,
- stopień uzyskania zadanej funkcjonalności,
- styl implementacji (przejrzystość kodu źródłowego),
- możliwość rozbudowy.

Oczekiwana funkcjonalność:

- Realizacja kreatywnego pomysłu na inteligentne, autonomiczne i/lub interaktywne sterowanie pojazdem eksperymentalnym zbudowanym na bazie uniwersalnego podwozia „Magician Chassis” - szczegóły do ustalenia z osobą prowadzącą zajęcia.

13 Ciekawa demonstracja zastosowania platformy Raspberry Pi Pico

Wymagania technologiczne:

- platforma sprzętowa: Raspberry Pi Pico

14 System monitorowania zużycia wody

Wymagania technologiczne:

- platforma sprzętowa: ESP32

15 Prosty oscyloskop USB oparty o zestaw STM Nucleo-144

Wymagania technologiczne:

- platforma sprzętowa: STM32 Nucleo-144 F746ZG

Informacje pomocnicze:

STM32F7 workshop: 05.1 USB oscilloscope with STM32CubeMX - First idea and STM32CubeMX installation