

Ütemterv

Járműinformatika

Tárgy kódja: GEIAL34J-B

Szak: járműmérnöki alapszak (BSc) nappali tagozat.

Típusa: kötelező

Oktató, előadó: dr. Kovács Szilveszter, dr. Baksáné dr. Varga Erika

Tárgyfelelős: dr. Kovács Szilveszter

Félév: 2018/2019 tavasz

Hét	Elmélet	Gyakorlat
1.	Bevezetés. Járműinformatika.	<i>Laborhasználati rend tájékoztató. Felhasználói account adminisztráció.</i> Ismétlés: a C programozási nyelv (a Számítástechnika c. tárgy tananyagának áttekintése). Adattípusok, változók és konstansok, operátorok és kifejezések, utasítások.
2.	A jármű elektronikus rendszerei.	Algoritmus fogalma és leírási módszerei (mondatszerű, folyamatábra). Szekvenciális és szelekciós vezérlési szerkezeteket tartalmazó algoritmusok leírása és kódolása.
3.	Modell alapú járműinformatikai fejlesztés.	Iterációs vezérlési szerkezeteket tartalmazó algoritmusok leírása és kódolása.
4.	Járműinformatikai vezérlő és ellenőrző rendszerek. Diszkrét rendszerek.	Komplex algoritmusok leírása és kódolása.
5.	Járműinformatikai beágyazott rendszerek bevezetés.	Standard C függvények hívása. A header állományok szerepe. Saját függvények írása C nyelven.
6.	A központi vezérlőegység fő típusai, mikrokontrollerek, az ASIC, ASIP, DSP, CPLD, FPGA és SoC koncepció	Adatgyűjtés és feldolgozás. A C nyelv összetett adattípusai (tömb).
7.	A mikrokontrollerek általános felépítése, architektúrák, memóriák	Adatgyűjtés és feldolgozás. A C nyelv összetett adattípusai (struktúra). <i>Féléves feladat kiadása.</i>

8.	<i>Rektori / dékáni szünet.</i>	<i>Rektori / dékáni szünet.</i>
9.	Memóriakezelés beágyazott rendszerekben, statikus memórafoglalás, verem (stack) és halom alapú memóriakezelés.	Fájlszintű adattárolás, fájlkezelés C-ben.
10.	Az autóiiparban leggyakrabban alkalmazott kommunikációs protokollok bevezetés.	Véges automaták (FSA) működésének leírása és kódolása.
11.	Kommunikációs protokollok: CAN	Véges automaták (FSA) működésének leírása és kódolása.
12.	Kommunikációs protokollok: LIN, MOST, FLEXRAY.	Numerikus integrálás.
13.	Összefoglalás.	<i>Féléves feladat leadása</i>
14.	Évközi zárthelyi dolgozat.	<i>Féléves feladat leadása</i>

Kötelező irodalom

- Kovács Szilveszter honlapján található előadásjegyzet (www.iit.uni-miskolc.hu/~szkovacs)

Ajánlott irodalom

- Jörg Schäuuffele, Thomas Zurawka: Automotive software engineering: principles, processes, methods, and tools, ISBN-10 0-7680-1490-5
- http://moodle.autolab.uni-pannon.hu/Mecha_tananyag/autoipari_kommunikacios_rendszerek/index.html
- http://moodle.autolab.uni-pannon.hu/Mecha_tananyag/autoipari_beagyazott_rendszerek/index.html

A tárgy lezárásának módja:

- aláírás, vizsgajegy

Évközi számonkérés:

- évközi zárthelyi dolgozat, amely az utolsó tanulmányi hétre esik.

Az aláírás megszerzésének feltételei:

- Az ME SzMSz III. kötet 38§ (6) pontja alapján, ha a hallgató nem igazolt hiányzása a gyakorlatokon eléri a gyakorlatok darabszámának 50%-át, a tantárgy aláírása nem szerezhető meg.
- Az aláírás megszerzésének további feltétele a zárthelyi dolgozatok elégséges szintű megírása.

Vizsga formája:

- írásbeli és szóbeli

Az írásbeli rész legalább elégséges teljesítése után következik a szóbeli rész. Az írásbeli és szóbeli rész értékelése:

0%-50% :	elégtelen
51%-62% :	elégséges
63%-75% :	közepes
76%-88% :	jó
89%-100% :	jeles

Az eredő teljesítmény a $0.667 \cdot \text{írásbeli} + 0.333 \cdot \text{szóbeli}$ képlettel kerül meghatározásra, melyhez jegy a megadott táblázat szerint rendelődik.

Elégtelen írásbeli vagy elégtelen szóbeli elégtelen vizsgajegyet jelent. A szóbelin a megjelenés kötelező. Az a hallgató, aki az írásbeli részen részt vett, de a szóbelin nem, „Nem jelent meg” Neptun bejegyzést kap.

A vizsgáztató oktatónak – ellenőrzési célból – joga van az írásbeli dolgozat egyes kérdéseinek szóban való ismételt reprodukálását kérni a hallgatótól.

Általános rendelkezések:

Az ME SzMSz III. kötet 96§ alapján a tárgyakhoz kapcsolódó valamennyi számonkérési alkalomnál a nem engedélyezett segédeszközök használata (puskázás) vagy más munkájának sajátként történő feltüntetése (plagizálás) fegyelmi vétségnek minősül, mely tanulmányi szankciókat vagy fegyelmi eljárást von maga után.

Tanulmányi szankció az évközi számonkéréseknél a számonkérés sikertelen minősítése. A számonkérés ilyen esetekben nem pótolható.

Tanulmányi szankció a vizsgaidőszakban a vizsga elégtelen minősítése, és hogy ismételt vizsgát a hallgató a tanszék által kijelölt időpontban, kijelölt vizsgabizottság előtt, szóbeli vizsga formájában tehet.

A puskázás és/vagy plagizálás tényét a tanszék a hallgató tanulmányi ideje alatt nyilvántartja, és ismételt előfordulás esetén a ME SzMSz III. kötet 96§ által előírt fegyelmi eljárást kezdeményez.

Miskolc, 2019. február 14.

Dr. Kovács Szilveszter

A verzió

2019. május 9.

Név, tankör:

Neptun kód:

Járműinformatika zárthelyi feladat

1. Feladat:

Járműinformatikai szempontból egy jármű milyen négy főbb alrendszerre bontható?
(1 pont)

2. Feladat:

A modell alapú szoftver fejlesztés kapcsán mit értünk „software in the loop” tesztelésen? (1 pont)

3. Feladat:

A modell alapú szoftver fejlesztés folyamatában mi a különbség az „ellenőrzés” és az „igazolás” céljában? (1 pont)

4. Feladat:

A vezérlő és ellenőrző rendszerek kapcsán mit értünk azon, hogy egy rendszer „diszkrét idejű”? (1 pont)

5. Feladat:

Mi a lényeges különbség egy „beágyazott rendszer” és egy általános célú számítógép feladatai és felépítése között? (1 pont)

A verzió

6. Feladat:

Mi a lényeges különbség a csak olvasható (ROM), valamint az írható és olvasható memóriák működése és felhasználása között? (1 pont)

7. Feladat:

Járműinformatikai szempontból mi az előnye a kommunikációs buszok alkalmazásának az egységek közötti független kapcsolatok építéséhez képest? (1 pont)

8. Feladat:

Kommunikációs szempontból mi a lényeges különbség a buszon használt „recesszív” és „domináns” bit működése között? (Pl. I2C, CAN, LIN) (1 pont)

9. Feladat:

Egy járműinformatikai rendszerben tipikusan milyen célokra használják a LIN buszt? Mi ennek az oka? (1 pont)

10. Feladat:

Milyen bit fog megjelenni a CAN buszon, ha egy időben az egyik CAN mester „recesszív” egy másik pedig „domináns” bit küld? (1 pont)

A verzió

2019. május 28.

Név, tankör:

Neptun kód:

Járműinformatika vizsga feladat

1. Feladat:

Mi a különbség egy jármű esetében az „onboard” és az „offboard” kommunikáció között? (1 pont)

2. Feladat:

Járműinformatikai szempontból egy jármű milyen négy főbb alrendszerre bontható? (1 pont)

3. Feladat:

A modell alapú szoftver fejlesztés kapcsán mit értünk „software in the loop” tesztelésen? (1 pont)

4. Feladat:

A modell alapú szoftver fejlesztés kapcsán mit értünk „hardware in the loop” tesztelésen? (1 pont)

5. Feladat:

A modell alapú szoftver fejlesztés folyamatában mi a különbség az „ellenőrzés” és az „igazolás” céljában? (1 pont)

A verzió

6. Feladat:

Mi a lényeges különbség egy „beágyazott rendszer” és egy általános célú számítógép feladatai és felépítése között? (1 pont)

7. Feladat:

Az irányító és ellenőrző rendszerek kapcsán mit értünk azon, hogy egy jel „diszkrét idejű” és „diszkrét értékű”? (1 pont)

8. Feladat:

Mik a „nem felejtő” memóriák működésének főbb jellemzői? (1 pont)

9. Feladat:

Hogyan jelzi a LIN mester az üzenet küldésének kezdetét? (1 pont)

10. Feladat:

Milyen bit fog megjelenni a CAN buszon, ha egy időben az egyik CAN mester „recesszív” egy másik pedig „domináns” bit küld? (1 pont)

A verzió

2019. május 9.

Név, tankör:

Neptun kód:

Járműinformatika zárthelyi feladat

1. Feladat:

Járműinformatikai szempontból egy jármű milyen négy főbb alrendszerre bontható?
(1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE2.pdf>

2o.-

2. Feladat:

A modell alapú szoftver fejlesztés kapcsán mit értünk „software in the loop”
tesztelésen? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE4.pdf>

3o.-

3. Feladat:

A modell alapú szoftver fejlesztés folyamatában mi a különbség az „ellenőrzés” és az
„igazolás” céljában? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE4.pdf>

5o.-

4. Feladat:

A vezérlő és ellenőrző rendszerek kapcsán mit értünk azon, hogy egy rendszer
„diszkrét idejű”? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE5.pdf>

8o.-

5. Feladat:

Mi a lényeges különbség egy „beágyazott rendszer” és egy általános célú számítógép
feladatai és felépítése között? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE6.pdf>

5o.-

A verzió

6. Feladat:

Mi a lényeges különbség a csak olvasható (ROM), valamint az írható és olvasható memóriák működése és felhasználása között? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE6.pdf>

48o.-

7. Feladat:

Járműinformatikai szempontból mi az előnye a kommunikációs buszok alkalmazásának az egységek közötti független kapcsolatok építéséhez képest? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE7.pdf>

4-7o.-

8. Feladat:

Kommunikációs szempontból mi a lényeges különbség a buszon használt „recesszív” és „domináns” bit működése között? (Pl. I2C, CAN, LIN) (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE7.pdf>

23-26o.-

9. Feladat:

Egy járműinformatikai rendszerben tipikusan milyen célokra használják a LIN buszt? Mi ennek az oka? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE7.pdf>

67o.-

10. Feladat:

Milyen bit fog megjelenni a CAN buszon, ha egy időben az egyik CAN mester „recesszív” egy másik pedig „domináns” bit küld? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE7.pdf>

100-101o.-

A verzió

2019. május 28.

Név, tankör:

Neptun kód:

Járműinformatika vizsga feladat

1. Feladat:

Mi a különbség egy jármű esetében az „onboard” és az „offboard” kommunikáció között? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE1.pdf>

10o.-

2. Feladat:

Járműinformatikai szempontból egy jármű milyen négy főbb alrendszerre bontható? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE2.pdf>

2o.-

3. Feladat:

A modell alapú szoftver fejlesztés kapcsán mit értünk „software in the loop” tesztelésen? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE4.pdf>

3o.-

4. Feladat:

A modell alapú szoftver fejlesztés kapcsán mit értünk „hardware in the loop” tesztelésen? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE4.pdf>

3o.-

5. Feladat:

A modell alapú szoftver fejlesztés folyamatában mi a különbség az „ellenőrzés” és az „igazolás” céljában? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE4.pdf>

5o.-

A verzió

6. Feladat:

Mi a lényeges különbség egy „beágyazott rendszer” és egy általános célú számítógép feladatai és felépítése között? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE6.pdf>

5o.-

7. Feladat:

Az irányító és ellenőrző rendszerek kapcsán mit értünk azon, hogy egy jel „diszkrét idejű” és „diszkrét értékű”? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE5.pdf>

8o.-

8. Feladat:

Mik a „nem felejtő” memóriák működésének főbb jellemzői? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE6.pdf>

48o.-

9. Feladat:

Hogyan jelzi a LIN mester az üzenet küldésének kezdetét? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE7.pdf>

74o.-

10. Feladat:

Milyen bit fog megjelenni a CAN buszon, ha egy időben az egyik CAN mester „recesszív” egy másik pedig „domináns” bit küld? (1 pont)

Lásd:

<https://www.iit.uni-miskolc.hu/~szkovacs/Jarmuinformatika/JarmuInfE7.pdf>

100-101o.-