FIŞA DISCIPLINEI

1. Date despre program

1.1 Instituţia de învăţământ superior	Universitatea Politehnica Timișoara
1.2 Facultatea[1] / Departamentul[2]	Inginerie Hunedoara/Inginerie Electrică și Informatică Industrială
1.3 Domeniul de studii (denumire/cod [3])	Inginerie electrică / 90
1.4 Ciclul de studii	Licență
1.5 Programul de studii (denumire/cod/calificarea)	Inginerie electrică și calculatoare / 60 / Inginer

2. Date despre disciplină

2.1a Denumirea disciplinei/Categoria formativă[4]			Programare în timp real / DS				
2.1b Denumirea disciplinei în limba engleză			Real-time programming				
2.2 Titularul activităţilor de curs			Ș. L. dr. ing. Abrudean Cristian				
2.3 Titularul activităţilor aplicative[5]			Ș. L. dr. ing. Abrudean Cristian				
2.4 Anul de studii[6]	IV	**2.5** Semestrul	8	**2.6** Tipul de evaluare	E	**2.7** Regimul disciplinei[7]	DOP

3. Timp total estimat - ore pe semestru: activități didactice directe (asistate integral sau asistate parțial) și activități de pregătire individuală (neasistate) [8]

3.1 Număr de ore asistate integral/săptămână	4 , format din:	**3.2** ore curs	2	**3.3** ore seminar/laborator/proiect	2
3.1* Număr total de ore asistate integral/sem.	56 , format din:	**3.2*** ore curs	28	**3.3*** ore seminar/laborator/proiect	28
3.4 Număr de ore asistate parțial/săptămână	, format din:	**3.5** ore practică		**3.6** ore elaborare proiect de diplomă	
3.4* Număr total de ore asistate parțial/ semestru	, format din:	**3.5*** ore practică		**3.6*** ore elaborare proiect de diplomă	
3.7 Număr de ore activități neasistate/ săptămână	3,14 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			1
		ore studiu individual după manual, suport de curs, bibliografie și notițe			1
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			1,14
3.7* Număr total de ore activități neasistate/ semestru	44 , format din:	ore documentare suplimentară în bibliotecă, pe platformele electronice de specialitate și pe teren			14
		ore studiu individual după manual, suport de curs, bibliografie și notițe			14
		ore pregătire seminarii/laboratoare, elaborare teme de casă și referate, portofolii și eseuri			16
3.8 Total ore/săptămână [9]	7,14				
3.8* Total ore/semestru	100				
3.9 Număr de credite	4				

4. Precondiţii (acolo unde este cazul)

4.1 de curriculum	• Programarea calculatoarelor. Sisteme de operare
4.2 de rezultatele învățării	•

5. Condiţii (acolo unde este cazul)

5.1 de desfăşurare a cursului	• Sală de curs proiectată cu videoproiector • Studenții nu se vor prezenta la prelegeri cu telefoanele mobile deschise • Nu se acceptă părăsirea sălii de curs fără aprobarea cadrului didactic
5.2 de desfăşurare a activităţilor practice	• Sală de laborator echipată cu computere • Studenții nu se vor prezenta la activitățile practice cu telefoanele mobile deschise • Nu se acceptă părăsirea sălii de de desfășurare a activității practice fără aprobarea cadrului didactic

6. Rezultatele învățării la formarea cărora contribuie disciplina

Cunoștințe	• C3. Studentul/absolventul explică și interpretează desenele care detaliază proiectarea produselor, a instrumentelor și a sistemelor de inginerie electrică.
Abilități	• A3. Studentul/absolventul specifică proprietăți tehnice ale bunurilor, materialelor, metodelor, proceselor, serviciilor, sistemelor, software-ului și funcționalităților, prin identificarea și răspunsul la nevoile particulare care urmează să fie satisfăcute în funcție de cerințele clienților. • A7. Utilizează eficient instrumente digitale de colaborare (MS Teams, Google Workspace, Moodle etc.) pentru învățare și lucru în echipă. • A8. Elaborează documentații tehnice, rapoarte și prezentări clare și corect structurate, în limba română și engleză.
Responsabilitate și autonomie	• RA2. Studentul/absolventul efectuează căutări bibliografice în literatura de specialitate, consultă și folosește bazele de date științifice și alte surse de informare din domeniul ingineriei electrice. • RA7. Își evaluează constant propriile nevoi de dezvoltare profesională și adoptă o atitudine proactivă față de învățarea pe tot parcursul vieții.

7. Obiectivele disciplinei (asociate rezultatelor învățării de la punctul 6)

- Obiectivele cursului constau în însușirea de către studenți a cunoștințelor teoretice aplicative a disciplinei de programare in timp real, cunostinte necesare pentru proiectarea si implementarea de aplicatii in timp real in sistemele de operare timp real. Insusirea acestei discipline are ca rezultat o pregătire de specialitate a studenţilor punandu-le la dispozitie cunostinte din domeniul programarii nesecventiale, programarii thredurilor si proceselor concurente in sistemele de operare de timp real, in special sistemul de operare QNX, cu ajutorul carora sa se poată alinia la progresul stiintei, sa-si dezvolte abilitati de programare cu constrangeri de timp real sub diverse familii de sisteme de operare real time(QNX, Real Time Linux); sa devina competenti pentru programarea de aplicatii real time
- Scopul formativ al cursului este ca studentul sa isi formeze o viziune de ansamblu asupra sistemelor integrate de timp-real. Se abordeaza in acest curs problematica sistemelor integrate de timp-real, din perspectiva dezvoltarii de aplicatii software de timp-real. Sunt exemplificate practic modalitatile de gestiune a timpului si de concurenta in dezvoltarea software-ului de timp-real. Dezvoltare de aplicatii in C/POSIX (QNX), Java si Real-Time Java (Windows / QNX / Linux). La finele cursului, studentii trebuie sa aiba cunostinte teoretice si abilitati de cercetare, strict necesare viitorilor specialisti, dovedind competente in selectarea, utilizarea corecta si combinarea adecvata a tehnologiilor de programare real/time in sistemele de operare de timp real.

8. Conţinuturi[10]

8.1 Curs	Număr de ore	Metode de predare[11]
1. Notiuni introductive despre sistemele integrate timp real 1.1. Sistem integrat de timp-real (embedded real-time system). Sisteme de operare de timp real. Arhitecturi software pentru proiectarea aplicatiilor de timp-real(executiv de tip ciclic, sisteme event-based, etc). 1.2. Limbaje de programare pentru aplicatii de timp-real (secventiale, concurentiale). Programare concurenta. Definitii (proces, fir de executie, task, proces, preemptiunea taskurilor). Probleme de baza ale proiectarii aplicatiilor de timp-real (sincronizare, excludere mutuala, deadlock, starvation).	4	Studenții au acces la curs în format electronic. Se vor utiliza atât prezentări interactive cât și tradiționale. Se vor folosi: problematizarea, studiu de caz, conversația.
2.1. Sincronizarea taskurilor. Sistem de activitati. Mecanisme pentru realizarea sincronizarii. Semafoare binare. 2.2. Semafoare generalizate. Excludere mutuala. Transmitere de mesaje. Cutii postale. 2.3. Sincronizarea cu ajutorul cutiilor postale. Excluderea mutuala cu ajutorul cutiilor postale.	8	

2.4. Buffere dinamice (pipes). Monitoare. Sincronizarea taskurilor cu ajutorul monitoarelor		
3. Aspecte temporale ale aplicatiilor de timp-real 3.1. Notiuni referitoare la "timp". Aspecte de timp-real (precizia masuratorilor, consistenta, curba functiei de utilitate). Utilizarea notiunii de "timp" intr-un sistem de timp-real. Scopuri temporale. 3.2. Caracteristici de limbaj de programare - timeout-uri, timere, alarme, semnale. Deadline-uri. Specificarea si analiza deadline-urilor. Prioritati. Algoritmi de evitare a inversiunii de prioritate. 3.3. Algoritmi de planificare (rate monotonic scheduling, deadline monotonic scheduling, arbitrary-fixed priority scheduling, earliest-deadline-first scheduling)..	6	
4.1. Organigrame, diagrame structurate, masini de stare finita, retele Petri, etc. 4.2. Metode de proiectare structurata (HRT-HOOD, UML pentru real-time). Ciclul de viata al software-ului. Standarde.	4	
5. Caracteristicile sistemelor de timp-real 5.1. Determinism, siguranta in functionare, etc. Planificarea taskurilor in sisteme de timp-real. Arhitecturi software pentru proiectarea aplicatiilor de timp-real. 5.2. Limbaje de programare pentru aplicatii de timp-real. Deadlock, starvation, cozi ciclice - algoritmi de detectie, evitare si prevenire. Interactiunea cu dispozitivele de I/O - mecanisme hardware I/O. Caracteristici de limbaj pentru programare low-level.	4	
6. Modele de stare si programe Java 6.1. Modelarea proceselor cu ajutorul masinilor de stare finita. 6.2. Procese si fire de executie.	2	

Bibliografie[12]

1. Abrudean C. – material didactic pe campus virtual UPT, https://cv.upt.ro/course/view.php?id=2719
2. Dragoicea, M., Programarea aplicatiilor in timp real. Editura Universitara, Bucuresti, 2009.
3. BOIAN F.M., FERDEAN C. M., BOIAN R.F. DRAGOS R.C. Programare concurenta pe platforme Unix, Windows, Editura Albastra - grupul Microinformatica, Cluj, 2002.
4. Wellings, A., Concurrent and Real-Time Programming in Java, John Wiley, 2004
5. Robu Nicolae, Programare concurentă: Mecanisme suport orientate timp real, Ed. Politehnica, 2002.

8.2 Activităţi aplicative[13]	Număr de ore	Metode de predare
1. Prezentarea sistemului de operare de timp-real QNX. Dezvoltarea aplicațiilor în QNX.	4	Verificare cunostinte din tematica laboratorului Elaborare de aplicatii interactive si testarea lor
2. Standardul POSIX pentru timp real. Mecanisme de sincronizare si comunicare sub QNX/POSIX. Semafoare. Sincronizarea firelor de executie. Sincronizarea proceselor	8	
3. Mecanisme de sincronizare si comunicare sub QNX / POSIX. Mutex-uri. Zone partajate de memorie. Variabile conditionale	4	
4. Mecanisme de sincronizare si comunicare sub QNX / POSIX. Buffere dinamice (pipes). Mesaje. Semnale.	4	
5. Planificarea taskurilor pe conditie de timp sub QNX / POSIX. Alarme. Timere	4	Verificare cunostinte din tematica laboratorului Elaborare de aplicatii interactive si testarea lor
6. Prezentarea pachetului Real-Time Java. Metode proiectare a aplicatiilor de timp-real. Masini de stare finita si programe Java	4	Verificare cunostinte din tematica laboratorului Elaborare de aplicatii interactive si testarea lor

Bibliografie[14]

1. Abrudean C. – material didactic pe campus virtual UPT, https://cv.upt.ro/course/view.php?id=2719
2. Dragoicea, M., Programarea aplicatiilor in timp real. Editura Universitara, Bucuresti, 2009.
3. BOIAN F.M., FERDEAN C. M., BOIAN R.F. DRAGOS R.C. Programare concurenta pe platforme Unix, Windows, Editura Albastra - grupul Microinformatica, Cluj, 2002.
4. Wellings, A., Concurrent and Real-Time Programming in Java, John Wiley, 2004

9. Evaluare

Tip activitate	**9.1** Criterii de evaluare[15]	**9.2** Metode de evaluare	**9.3** Pondere din nota finală
9.4 Curs	4 subiecte cu caracter aplicativ care demonstrează însușirea elementelor prezentate la curs	Examen pe calculator	0,66
9.5 Activităţi aplicative	**S:**		
	L: Lucrări de control, teme de casă și răspunsurile la întrebările puse la laborator	Lucrări de control pe calculator, teme pe suport electronic, conversație	0,34
	P[16]**:**		
	Pr:		
9.6 Standard minim de performanţă (se prezintă cunoștințele minim necesare pentru promovarea disciplinei și modul în care se verifică stăpânirea lor[17])			
• Nota de promovare se obține în condițiile obținerii a cel puțin jumătate din punctajul total cu condiția implementării a cel puțin două programe, prin implementare înțelegând cel puțin că trebuie să se compileze corect. Nota finală constă în media ponderată a notei de la examen și a celei de la activitatea aplicativă (laborator). Nota maximă se acordă pentru rezolvarea în totalitate a subiectelor și minim nota 9 la activitatea pe parcurs.			

Data completării	**Titular de curs (semnătura)**	**Titular activităţi aplicative (semnătura)**
10.09.2026		
Director de departament (semnătura)	**Data avizării în Consiliul Facultăţii**[18]	**Decan (semnătura)**
	23.09.2026	