

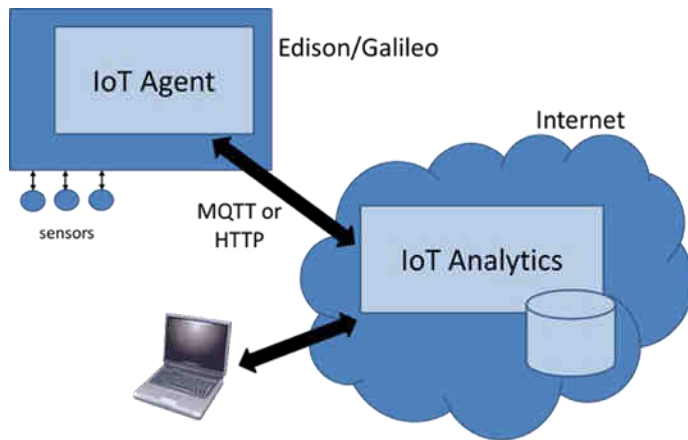


UNIVERSITATEA TEHNICĂ A MOLDOVEI

Digitally signed by
Technical Scientific
Library, TUM
Reason: I attest to the
accuracy and integrity of
this document

SISTEME DEDICATE DE CONTROL

*Îndrumar de laborator
și elaborare a lucrării individuale*



**Chișinău
2022**

UNIVERSITATEA TEHNICĂ A MOLDOVEI

**FACULTATEA CALCULATOARE, INFORMATICĂ
ȘI MICROELECTRONICĂ
DEPARTAMENTUL INFORMATICĂ
ȘI INGINERIA SISTEMELOR**

**Victor ABABII
Viorica SUDACEVSCHI
Silvia MUNTEANU**

SISTEME DEDICATE DE CONTROL

*Îndrumar de laborator
și elaborare a lucrării individuale*

**Chișinău
Editura "Tehnică-UTM"
2022**

CZU

Îndrumarul de laborator și elaborare a lucrării individuale este destinat masteranzilor anului I, semestrul 2, programul de studii *Calculatoare și rețele informaționale*, forma de studii cu frecvență la zi, Facultatea Calculatoare, Informatică și Microelectronică.

Îndrumarul include material teoretic, practic și variante individuale și de grup pentru efectuarea a patru lucrări de laborator și elaborarea unei lucrări individuale. Lucrarea individuală prevede integrarea cunoștințelor acumulate în procesul elaborării lucrărilor de laborator pentru proiectarea și programarea unui sistem dedicat de control integrat în IoT.

Obiectivele îndrumarului sunt consolidarea materialului teoretic predat la orele de curs și analiza exemplurilor concrete de proiectare a sistemelor dedicate de control bazate pe modele de inteligență artificială.

Autori: conf. univ., dr. Victor Ababii
conf. univ., dr. Viorica Sudacevschi
asist. univ. Silvia Munteanu

Redactor responsabil: prof. univ., dr. Vasile Moraru
Recenzent: prof. univ., dr. hab. Emilian Guțuleac

INTRODUCERE	4
DESCRIEREA PARAMETRICĂ ȘI FUNCȚIONALĂ A DISPOZITIVELOR	10
DESCRIEREA MEDIULUI DE DEZVOLTARE	14
LUCRAREA DE LABORATOR NR. 1	17
Noțiuni teoretice	18
Formularea sarcinii de proiectare	22
Variante pentru îndeplinirea lucrării de laborator	25
Întrebări pentru verificare	27
LUCRAREA DE LABORATOR NR. 2	28
Noțiuni teoretice	29
Formularea sarcinii de proiectare	35
Lista de variante	35
Lista de întrebări pentru verificare	37
LUCRAREA DE LABORATOR NR. 3	38
Noțiuni teoretice	39
Formularea sarcinii de proiectare	44
Lista de variante	44
Lista de întrebări pentru verificare	46
LUCRAREA DE LABORATOR NR. 4	47
Noțiuni teoretice	48
Formularea sarcinii de proiectare	53
Lista de întrebări pentru verificare	54
LUCRARE INDIVIDUALĂ	55
BIBLIOGRAFIE	59

INTRODUCERE

Disciplina *Sisteme dedicate de control* face parte din programul de studii superioare de master, Ciclul II, *Calculatoare și rețele informaționale*. Este disciplină de specializare predată în semestrul II, cu o pondere de 5 credite ECTS [1].

La baza disciplinei stau cunoștințele acumulate de masteranzi pe parcursul anilor de studii în cadrul programului de licență (*Matematica superioară, Matematica discretă, Mecanica teoretică, Fizica, Programarea calculatoarelor, Structuri de date și algoritmi, Bazele transiterii de date, Arhitectura calculatoarelor, Proiectarea sistemelor cu microprocesor, Inteligența artificială* etc.) și studii în cadrul programului de master la disciplinele: *Comunicații mobile și prin satelit, Proiect IoT, Modelarea și identificarea sistemelor tehnice, Verificarea și evaluarea performanțelor sistemelor orientate obiect, Arhitecturi Cloud, Programarea dispozitivelor mobile, Securitatea avansată a rețelelor informatice și Aplicații Cloud*.

Conținutul curricular al disciplinei *Sisteme dedicate de control* este orientat spre dezvoltarea competențelor și abilităților necesare unui inginer, dezvoltator de sisteme/aplicații și cercetător, precum:

- identificarea și descrierea formală și tehnologică a metodelor de rezolvare a problemelor complexe bazate pe modelare, simulare, verificare și implementare a sistemelor dedicate de control bazate pe Inteligența Artificială;
- aplicarea cunoștințelor interdisciplinare, a metodelor de soluționare și a mediilor de dezvoltare și testare funcțională a sistemelor dedicate de control;
- aplicarea metodelor și tehnicilor de soluționare a problemelor din diverse domenii ale economiei naționale, utilizând medii de proiectare asistată de calculator: CAD, CAE, CASE etc.;
- analiza și evaluarea comparativă a performanțelor sistemelor de calcul și ale aplicațiilor software, utilizând metode și modele alternative în scopul optimizării structurale și algoritmice;

- dezvoltarea abilităților de lucru individual și în echipă.

Sistemele dedicate de control reprezintă o combinație de structuri de calcul hardware și software, circuite electronice și respectiv componente electromecanice, concepute pentru a realiza o funcție specifică (dedicată unei aplicații) și include funcții de achiziție, procesare și acțiune.

La nivelul structurii hardware a sistemului dedicat de control stau componentele standard ale sistemelor de calcul: microprocesoare (MPU) sau microcontrolere (MCU) universale sau specializate, procesoare de semnal DSP. Aceste arhitecturi asigură implementarea algoritmilor de control în formă de produse program, care realizează logica funcțională.

Paralel cu arhitecturile clasice bazate pe MPU și MCU, în sistemele dedicate de control sunt utilizate și implementări tehnologice hardware:

- Application-Specific Integrated Circuits (ASIC);
- Application-Specific Standard Products (ASSP);
- Field Programmable Gate Array (FPGA);
- Soluția “System on Chip” (SOC).

La nivelul structurii software a sistemului dedicat de control stau: sistemul de operare, medii de dezvoltare, limbajele de programare, drivere și biblioteci de module sau produse program.

Sistemul de operare (SO) este un set de programe de calcul care controlează resursele hardware și software ale sistemului dedicat de control. Pentru a se asigura o concurență corectă, aceste produse program trebuie să fie dedicate (embedded) și să funcționeze în timp real (Real-time Operating System - RTOS).

Un sistem de operare dedicat se caracterizează prin următoarele:

- dimensiuni mici (uneori sute de octeți);
- trebuie să fie capabil să ruleze pe perioade mari de timp (ani de zile) fără intervenție manuală.

Sarcinile unui sistem de operare pentru sisteme dedicate de control sunt:

- planificarea proceselor (task-urilor);
- deservirea întreruperilor;
- comunicarea și sincronizarea între procese;
- managementul memoriei;
- administrarea sistemului de fișiere;
- asigurarea conexiunii în rețeaua de calculatoare (TCP/IP);
- comanda Interfaței Grafice-Utilizator (Graphical User Interface -GUI).

Un sistem de operare în timp real este capabil să execute toate sarcinile sale, respectând anumite constrângeri de timp bine definite.

Menționăm câteva caracteristici specifice SO pentru sistemele dedicate de control:

- răspuns rapid și predictibil la evenimentele externe urgente;
- comportament stabil la supraîncărcări tranzitorii (când sistemul este supraîncărcat de evenimente astfel încât nu pot fi respectate toate constrângerile de timp impuse pentru anumite procese critice, cu respectarea timpului de execuție);
- un înalt grad de programabilitate (constrângerile de timp ale sistemului trebuie să fie satisfăcute la un nivel ridicat de utilizare a tuturor resurselor);
- trecerea în regimurile de consum redus de energie se face setând corespunzător anumiți biți dintr-un registru de comandă (PCON).

Medii de dezvoltare IDE

Un mediu integrat de dezvoltare (Integrated Development Environment (IDE) sau Integrated Design Environment and Integrated Debugging (IDEID) este o aplicație software care asistă programatorul la dezvoltarea programelor. Un IDE include de regulă:

- editor de cod sursă;
- asamblor/compiler/interpretor;
- depanator (debugger);

- unelte pentru elaborarea interfeței grafice-utilizator (GUI).

Editorul de cod sursă are capabilități de “highlighting” și autocompletare. Este un editor de text “non document”.

Compilerul este o aplicație software care translatează textul scris într-un limbaj de programare (cod sursă) în alt limbaj de programare (cod obiect). Compilerul este utilizat pentru a transla un program sursă scris în limbaj de nivel înalt (C, C++, etc.), într-un cod cu limbaj de nivel mai scăzut (limbaj de asamblare) sau limbaj mașină. Limbajul de asamblare este un limbaj de nivel scăzut orientat spre arhitectura MPU sau MCU. Rezultatul asamblării îl constituie codul obiect (care se execută).

Companiile producătoare de MPU și MCU împreună cu companiile producătoare de compilatoare și IDE oferă o gamă foarte largă de medii de dezvoltare care combină facilitățile de administrare a proiectului, editare a codului sursă și depanare a programelor într-un singur mediu performant. Complexitatea acestor medii de dezvoltare poate fi explicată prin analiza componentelor care fac parte din IDE “μVision” utilizat pentru dezvoltarea produselor program în bază de MCU Intel/AVR:

- **C51 ANSI C** – compilator optimizat (C Cross Compiler) care creează din codul sursă C module obiect relocabile;
- **A51 Macro Assembler** – creează module relocabile din codul sursă în limbaj de asamblare;
- **BL51 Linker/Locator** – combină modulele obiect relocabile create de C51 și A51 într-un modul obiect absolut;
- **LIB51** – bibliotecarul (Library Manager) combină modulele obiect într-o bibliotecă ce poate fi utilizată de linker (editorul de legături);
- **OH51** – convertor cod obiect-HEX, creează fișiere Intel HEX din module cod absolut;
- **RTX-51** – sistem de operare în timp real (RTOS) care simplifică elaborarea aplicațiilor software complexe pentru sistemele dedicate de control.

Emularea hardware este procesul de imitare a comportamentului uneia sau mai multor componente hardware, incluse în sistemul proiectat, cu o altă componentă hardware, numită sistem de emulare sau emulator. Acesta se implementează prin interfețe specifice incluse în arhitectura chip-ului, de exemplu BDM (Background Debug Mode) care utilizează interfața JTAG pentru a testa funcționalitatea Unității Centrale de Procesare (Central Processing Unit - CPU) a microcontrolerului.

Limbajul de asamblare constituie o reprezentare simbolică a codului mașină și a constantelor necesare pentru a programa un CPU. Arhitectural, codul mașină reprezintă un sistem de instrucțiuni și date ce pot fi direct interpretate și executate de către CPU.

Limbajul de asamblare este definit, de regulă, de fabricantul hardware-ului și se bazează pe abrevieri sugestive (mnemonice) care sugerează efectul instrucțiunii, registrele implicate etc.

Limbajul de asamblare este specific unei arhitecturi fizice sau virtuale a CPU și nu este portabil de la o arhitectură la alta. Poate fi specificată doar o compatibilitate logică sau funcțională dintre două arhitecturi și limbajele de asamblare ale acestora.

Pot fi menționate câteva caracteristici specifice limbajelor de asamblare:

- programarea în limbaj de asamblare este mai dificilă (necesită cunoașterea arhitecturii hardware, risc mare de apariție a erorilor, productivitate scăzută etc.);
- programul scris în limbajul de asamblare conduce la un cod mașină optimal (dimensiuni mici și viteză mare de execuție);
- programul scris în limbajul de asamblare este utilizat pentru comanda directă a subansamblului hardware pentru porțiuni de program care necesită viteză sporită de execuție, memorie program de mică capacitate (drivere pentru dispozitive hardware, sisteme dedicate, sisteme de timp real etc.).

Asamblorul este un program utilitar care translatează instrucțiunile din limbajul de asamblare în instrucțiuni scrise în cod mașină specifice CPU ale MPU/MCU utilizat. Procesul de traducere

este izomorf (unu-la-unu) de la mnemonice la instrucțiuni și date cod mașină. Majoritatea programelor de asamblare (numite macroasamblare) suportă macrouri, adică combinații de simboluri care substituie o secvență de linii program (instrucțiuni, date). Odată definit ca pseudocod, numele macroului poate fi utilizat în locul codului operațional pe care îl echivalează.

BIBLIOGRAFIE

1. https://utm.md/wp-content/uploads/2021/06/Plan_FCIM_CRI-modif-18.05.2021.pdf (Accesat 15.10.2021).
2. <https://habr.com/ru/post/236701/> (Accesat 25.10.2021).
3. <https://arduino.cc/> (Accesat 27.10.2021).
4. https://wiki.dfrobot.com/Romeo_for_Edison_Controller_SK_U_DFR0331 (Accesat 27.10.2021).
5. <https://ru.mouser.com/new/dfrobot/dfrobot-romeo-intel-edison/> (Accesat 02.11.2021).
6. <http://www.ace.tuiasi.ro/users/103/2009-Dosoftei%20Catalin%20PhD%202009.pdf> (Accesat 12.11.2021).
7. <http://www.bel.utcluj.ro/dce/didactic> (Accesat 13.11.2021).
8. https://staff.fmi.uvt.ro/~daniela.zaharie/am2016/curs/curs12/am2016_slides12_RN.pdf (Accesat 13.11.2021).
9. <https://ru.scribd.com/doc/44518986/Carte-Retele-Neuronale> (Accesat 13.11.2021).
10. https://www.researchgate.net/publication/24124504_Algoritmi_genetici (Accesat 20.11.2021).
11. <https://webspace.ulbsibiu.ro/daniel.morariu/html/StudentDoc/ML/IA-laborator5.pdf> (Accesat 25.11.2021).
12. https://www.researchgate.net/publication/338800538_Internet_of_Things (Accesat 29.11.2021).
13. https://www.researchgate.net/publication/352755798_Internet_of_Things_in_energie_tehnologii_aplicatii_provocari_si_tendinte_viitoare (Accesat 30.11.2021).