

Universitatea Tehnică “Gheorghe Asachi” din Iași

Facultatea de Automatică și Calculatoare

Domeniul: Ingineria sistemelor

Specializarea: Automatică și Informatică Aplicată

Anul universitar 2023-2024

Proiect la disciplina

Comunicații în sisteme de conducere

Tema 4

Antici Paul-Leonard

1. Tema Proiectului

Folosind microsistemul BIG8051 să se implementeze un protocol de comunicație serială pentru 5 noduri conectate în rețea, având următoarele caracteristici:

1	Port	UART1 (RS-485 folosind un adaptor extern TTL – RS -485
2	Parametri comunicație	115200, 9, N, 1
3	Formatul mesajului	Adresa HW nod destinație
		Adresa HW nod sursa
		Tipul mesajului: 0 sau 1
		Sursa mesajului (numai mesaje de tip 1)
		Destinația finală a mesajului (numai mesaje de tip 1)
		Lungime date (numai mesaje de tip 1)
		Date (numai mesaje de tip 1)
		Cod detectare erori: LRC
4	Codificare mesaj	Binar

Formatul mesajelor:

Adresă hardware nod destinație	Adresă hardware nod sursă	Tip mesaj	Adresă nod sursă mesaj	Adresă nod destinație mesaj	Lungime	Date	Suma de control
1 octet	1 octet	1 octet	1 octet	1 octet	1 octet	Lungime	1 octet

Observatii:

- Câmpul adresă hardware nod destinație va conține adresa nodului care trebuie să recepționeze mesajul.
- Câmpul adresă hardware nod sursă va conține adresa nodului care efectiv transmite mesajul.
- Câmpul tip mesaj poate fi, după caz:

- 0:
 - mesaj de interogare transmis de master către un nod slave sau de slave către master, în lipsa altui mesaj util;
 - mesaj de tip jeton, trimis de către un nod pentru transferul jetonului către un alt nod.
 - 1 – mesaj de date – mesaj care conține un text ASCII introdus de utilizator de la interfața de intrare a unui nod.
- Câmpul adresă nod sursă mesaj va indica sursa mesajului;
 - Câmpul adresă nod destinație mesaj va indica destinația finală a mesajului:
 - un mesaj de tip 1, transmis de un nod slave către un alt node slave, va fi retransmis de către nodul master către nodul slave destinație finală;
 - la protocolul bazat pe jeton poate lipsi (este identic cu adresa destinație hardware).
 - Câmpul lungime reprezintă numărul de octeți al câmpului de date;
 - Octetul din câmpul suma de control va fi calculat ca suma modulo 2 a tuturor octeților din câmpurile anterioare.
 - Mesajul va fi codificat în binar și transmis octet cu octet portului serial UART0, conectat la un adaptor extern TTL - RS-485.

2. Prezentarea resurselor HW și SW utilizate

Ca și componentă HW am folosit:

- 1) Microsistemul BIG8051 – MikroElektronika

Acesta oferă o platformă flexibilă pentru programarea și dezvoltarea aplicațiilor folosind microcontrolere 8051.

Un număr mare de module precum butoane, LED-uri, afișaje LCD, difuzoare piezoelectrice (buzzers), interfețe de comunicație și controlere, convertoare analog-digital și digital-analogic permit utilizatorilor să testeze cu ușurință funcționalitatea aplicației direct pe PC.



Fig. 1.1 Microsistemul BIG8051

1. Modul interfață Ethernet
2. Buzzer piezoelectric
3. Conector pentru senzorul de temperatură
4. Conector pentru programator/debugger
5. Modul USB-UART și alimentare USB
6. Modul interfață RS-232 - A
7. Conector card MCU
8. Modul interfață RS-232 - B
9. Comparator
10. Jumperi selectare nivel de fixare VCC | GND

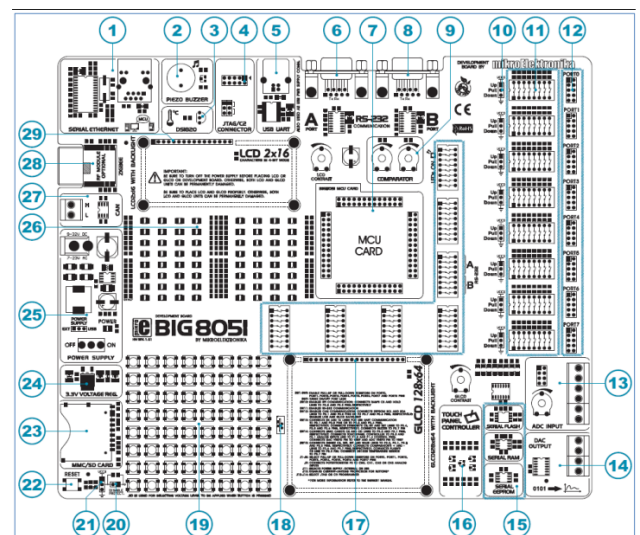


Fig. 1.2 BIG8051 – blocuri componente

11. DIP switch conectare pini la rezistoare de fixare
12. Conectori porturi I/E și alimentare
13. Intrări convertor A/D
14. Ieșiri convertor D/A
15. Modul memorii seriale externe
16. Controler touch panel
17. Conector pentru afișaj grafic (GLDC)
18. Conector pentru touch panel
19. Butoane cu revenire conectate la pinii porturilor

20. Jumper scurtcircuitare rezistor de protecție
 21. Jumper selectare stare logică buton apăsat
 22. Buton RESET
 23. Conector card de memorie MMC/SD
 24. Regulator de tensiune 3,3V c.c.

25. Conector și selecție sursă de alimentare
 26. LED-uri pentru afișarea stării porturilor
 27. Modul interfață CAN
 28. Conector interfață ZigBee
 29. Conector afișaj LCD alfanumeric

Sistemul de dezvoltare BIG8051 este echipat cu un microcontroler C8051F040 de la Silicon Laboratories, într-o capsulă TQFP cu 100 de pini, lipită pe un card MCU montat pe placă (Fig.1.3).

Caracteristici tehnice ale microcontrolerului C8051F040:

- microcontroler MSP (Mixed-Signal Processor) – procesare semnale digitale și analogice;
- compatibil software cu 8051 - același set de instrucțiuni;
- arhitectură pipeline - execută 70% din setul de instrucțiuni în 1 sau 2 perioade de tact;
- oscilator programabil intern, calibrat, $3 \div 24,5$ MHz;
- execută până la 25 MIPS cu oscilator de 25 MHz;
- RAM intern de date de 4352 octeți (4KB + 256);
- 64 KB memorie Flash, programabilă în sistem (ISP) în sectoare de 512 octeți;
- 8 porturi de I/E configurabile;
- set extins de numărătoare – temporizatoare (Timers & PCA – Programmable Counter Array)
- convertoare A/D și D/A
- set extins de periferice de comunicații:
- 2 porturi seriale UART și un port SPI;
- Bosch Controller Area Network (CAN 2.0B);
- SMBus (compatibil I2C).

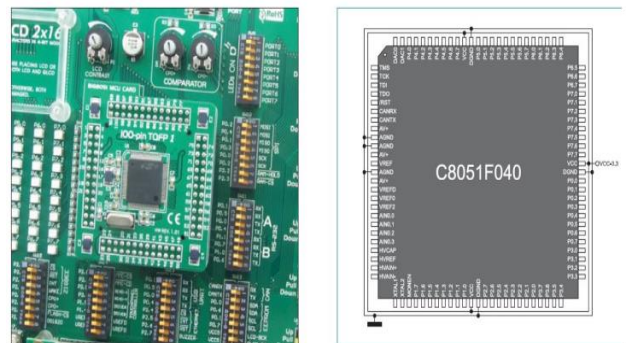


Fig. 1.3 Microcontrolerul C8051F040 și cardul MCU

Programarea microcontrolerului și depanarea programului:

Microcontrolerul C8051F040 este amplasat pe cardul MCU, care este montat pe sistemul de dezvoltare și poate fi programat cu un adaptor USB DEBUG de la Silicon Laboratories. Acesta se conectează ca în Fig.1.4 și funcționează atât ca debugger JTAG, cât și ca debugger C2. Pentru microcontrolerul C8051F040 se utilizează varianta JTAG, cu ajutorul J13 și J14.



Fig.1.4 Conectarea adaptorului USB DEBUG

Organizarea memoriei microcontrolerului C8051F040:

Aceasta este similară cu cea de la 8051 standard (arhitectura Harvard modificată). Există două spații de memorie separate: de program și de date (Fig.1.5). Ele partajează același spațiu de

adrese, dar sunt accesate cu tipuri diferite de instrucțiuni. CIP-51 are implementat un spațiu de 256 de adrese de memorie RAM internă și de 64KB de memorie internă program de tip Flash.

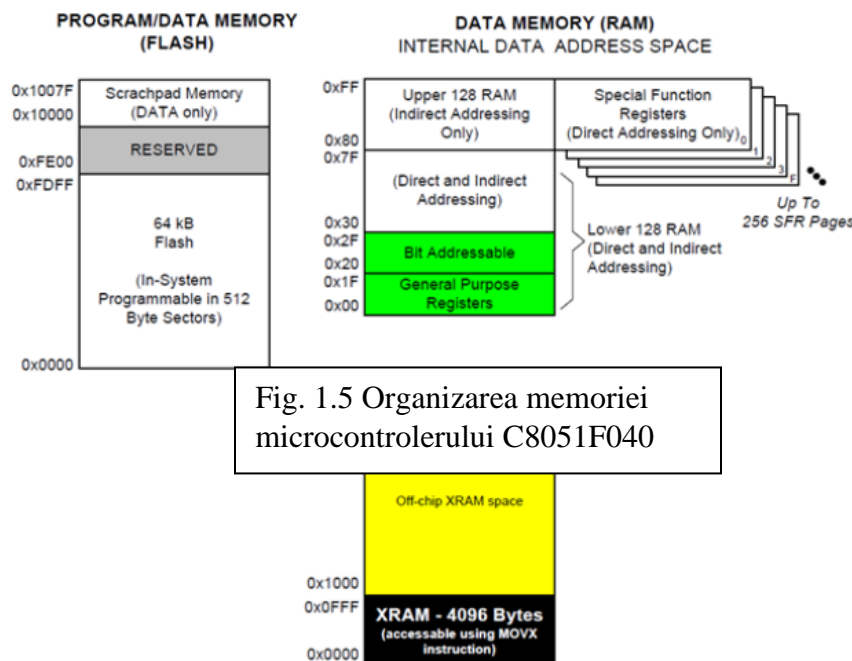


Fig. 1.5 Organizarea memoriei microcontrolerului C8051F040

Memoria program:

Microcontrolerul

C8051F040 are implementat un spațiu de 64Kb de memorie program,

sub forma unui bloc continuu 0000h-FFFFh. Ultimii 512 octeți (FE00h-FFFFh) sunt rezervați din fabrică și nu trebuie folosiți. Memoria program este înscrisă cu ajutorul programatorului JTAG și poate fi de regulă doar citită din programul utilizatorului:

- când sunt extrase instrucțiunile pentru a fi executate;
- cu ajutorul unor instrucțiuni MOVC, pentru acces la date constante, amplasate în memoria program.

Memoria de date:

Spațiul de memorie internă de date de 256 de octeți RAM este organizat la fel ca la 8052. Zona inferioară de 128 octeți este adresabilă atât direct, cât și indirect și aici regăsim cele 4 bancuri de registre de uz general (00h-1Fh), în care sunt mapate registrele R0-R7, precum și un spațiu de 16 locații adresabile atât pe octet, cât și pe bit (adrese de octet 20h-3Fh și adrese de bit 00h-7Fh).

Zona superioară de memorie, adresabilă numai indirect și utilizată de obicei pentru stivă, ocupă același spațiu de adrese cu zona registrelor cu funcții speciale (SFR), care este adresată direct. Diferitele spații de memorie ale microcontrolerelor din familia 8051 pot fi folosite pentru amplasarea codului și datelor folosind opțiuni ale mediului integrat de dezvoltare Keil μVision (pentru a stabili reguli implicite) sau extensii ale limbajului C pentru specificarea explicită a spațiului de memorie în care va fi amplasat codul și variabilele programului.

Astfel, codul aplicației este amplasat întotdeauna, implicit în memoria program, acest lucru nu mai trebuie specificat explicit.

Conectarea dintre placă și calculator:

Comunicația între cele 2 se realizează cu ajutorul programelor **uVision5 si Cypress Serial Port Viewer and Terminal**. Cu ajutorul acestora, vom putea compila și hex-edita programe ce vor fi încărcate ulterior pe placă cu ajutorul debuggerului.

uVision5 este un mediu de dezvoltare integrat (IDE) creat de compania Keil, specializat în dezvoltarea și depanarea codului pentru microcontrolerele bazate pe arhitecturi ARM. Acest program oferă un set cuprinzător de instrumente pentru scrierea, compilarea, depanarea și testarea codului încorporat. Cu ajutorul uVision5, programatorii pot crea și gestiona proiecte complexe pentru o varietate de dispozitive ARM, beneficiind de funcționalități precum editor de cod, compilator, simulator, analizator static, și depanator în timp real. Este un instrument puternic utilizat în industria electronică pentru dezvoltarea aplicațiilor încorporate.

3. Protocolul Master-Slave

Protocolul master-slave este un tip de arhitectură de comunicație în care un dispozitiv (numit master) controlează unul sau mai multe dispozitive (numite slave). Nodul master inițiază și gestionează comunicația cu nodurile slave, care răspund la cererile acestuia.

În cadrul acestui protocol, nodul master are rolul predominant și dictează fluxul de date. El inițiază comunicarea, trimite comenzi și solicită informații de la slaves. Nodurile slave, pe de altă parte, sunt dispozitive pasive care așteaptă comenzi de la master și răspund în consecință.

Comunicarea între master și slave poate fi de tip unidirecțională sau bidirecțională, în funcție de necesitățile sistemului. Masterul poate trimite comenzi și să solicite date de la slave-uri, iar aceștia pot răspunde cu informații sau să execute acțiuni în funcție de cerințele primite.

Protocolul master-slave este utilizat într-o varietate de aplicații, inclusiv în sistemele de comunicație seriale precum RS-232, RS-485, I2C, SPI și în rețelele de calculatoare unde un nod central (master) coordonează comunicația cu alte noduri periferice (slaves).

Descrierea functionarii programului pentru nodul master:

Secvența de inițializare

- afișaj LCD;
- coprocesor tastatură;
- port serial;
- variabilele programului (de exemplu: adresa HW a nodului).

Afișare parametri program:

Linia 1: Master/Slave, Adresa, COM0/COM2, ASCII/Binar.

Linia 2: RxM – nod - ultimul mesaj de date recepționat cu succes și care nod l-a transmis.

Afișare meniu comenzi:

Linia 3: 1 – TxM 2 – Stare.

Linia 4: utilizată de comenzile 1 și 2.

Masterul este cel care coordonează transmisia mesajelor în rețea. Alege un slave inițial (slave 1), apoi va intra în buclă și va urma setul următor de instrucțiuni.

Verifică dacă are un mesaj de transmis nodului respectiv. În cazul în care este un mesaj de trimis, masterul îl va trimite la adresa indicată prin funcția TxMesaj(nod destinație) ; în caz contrar trimite un mesaj de interogare (cod funcție 0).

Apelează funcția RxMesaj(slave) cu adresa nodului de la care se așteaptă un raspuns trimisă ca parametru (așteptarea se va face cu timeout o secundă , moment în care se va trece la următorul slave). În cazul în care mesajul receptat este de tip 0, se va trece la următorul nod slave. Pentru un mesaj de tip 1 adresat nodului master, acesta îl va afișa pe ecran (linia a doua a LCD-ului). Dacă vine un mesaj de tip 1 pentru destinat altui slave, îl va memora pentru o transmitere ulterioară.

După terminarea funcției RxMesaj se verifică dacă s-a apăsă o tasta. Pentru valoarea 1 (TxM) va solicita adresa și datele care urmează a fi trimise slave-ului , iar pentru valoarea 2 se va verifica și afișa starea bufferului nodului slave indicat.

Descrierea funcționării programului pentru nodul slave:

Nodul slave funcționează într-o buclă infinită, așteptând fie un mesaj de la master, fie o comandă de la tastatură. De la master, poate primi un mesaj de tip 0 sau de tip 1. În cazul în care primește un mesaj de tip 1, acesta este afișat pe linia 2 și transmis înapoi către master. Dacă nu are un mesaj de tip 1 de transmis, nodul slave trimite către master un mesaj de tip 0. În cazul apariției unei comenzi de la tastatură, aceasta este procesată în funcție de tipul său. Pentru comanda "TxM", se solicită adresa destinație, apoi mesajul este preluat și memorat. Pentru comanda "stare mesaje", se solicită adresa destinație și se afișează dacă există sau nu un mesaj așteptând să fie transmis.

4. Interfața I/E și conectarea la mediul de comunicație (nivelul fizic)

Microcontrolerul C8051F040 dispune de 8 porturi de I/E de 8 biți (P0-P7). Toate porturile sunt adresabile și pe bit, iar pinii porturilor suportă și nivele TTL (5V)

Toți pinii porturilor pot fi configurați în 3 moduri de funcționare:

- ☐ operare normală (push-pull);
- ☐ drenă în gol (ȘI cablat);
- ☐ cu fixare slabă a nivelului la VDD (weak pullup)

Operare normală (push – pull) – comanda pentru pin (0 sau 1) vine pe linia PORT-OUTPUT și dacă driverul de ieșire este validat (/PORT-OUTENABLE = 0), comandă cele două tranzistoare în opoziție, iar pe pin este forțat nivelul dorit: 0 sau 1.

✓ În acest mod, pinul este configurat ca ieșire și nu poate fi conectat la un alt pin de ieșire, ci doar la pini de intrare. Starea pinului de ieșire poate fi citită prin program, dacă pinul nu este selectat ca intrare analogică.

Operare cu drenă în gol – se obține dacă /PORT-OUTENABLE = 1: tranzistorul de sus este blocat continuu și cel de jos este deschis numai când comanda pe linia PORT-OUTPUT este 0. Astfel, microcontrolerul poate forța pinul doar pe nivel 0; când se comandă nivel 1 pinul flotează (HZ), dacă /WEAK-PULLUP = 1.

✓ În acest mod, pinul poate funcționa ca ieșire cu drenă în gol și poate fi conectat la o altă ieșire cu drenă în gol, precum și la pini de intrare. Starea pinului poate fi citită prin program dacă pinul nu este selectat ca intrare analogică. Acest mod este folosit pentru conectarea mai multor ieșiri împreună, la aceeași linie fizică.

✓ În acest mod pinul poate funcționa și ca intrare, cu condiția ca PORT-OUTPUT să fie fixat la 1.

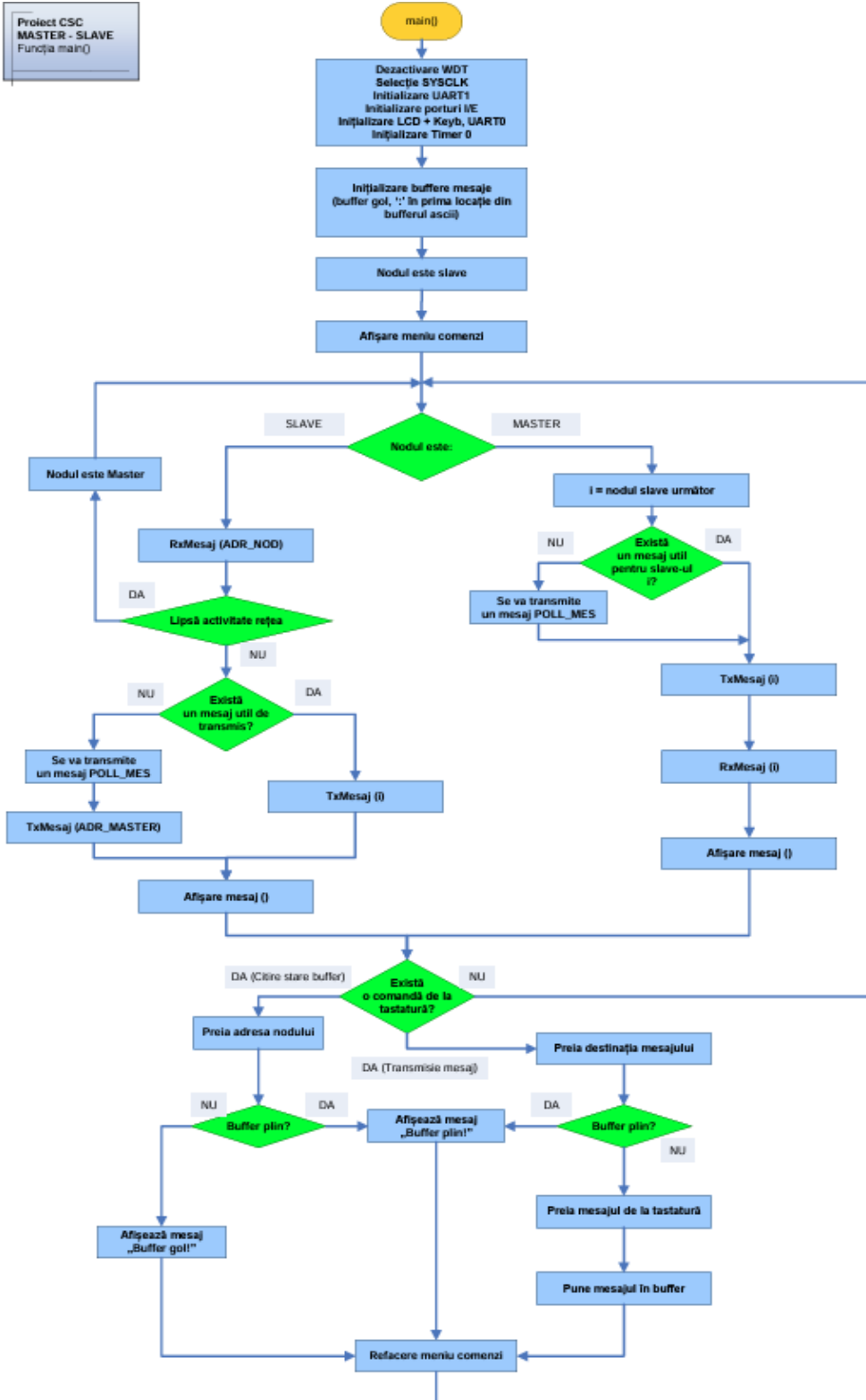
Operarea cu fixare slabă la VDD – când /PORT-OUTENABLE = 1, ambele tranzistoare sunt blocate și nivelul pe pin poate fi fixat slab la VDD prin /WEAK-PULLUP = 0 (se conectează printr-o rezistență internă de cca. 100KΩ la Vcc), dacă pinul nu este folosit ca intrare analogică.

✓ În acest mod starea pinului, care poate fi citită prin program, este 1 dacă pinul nu este

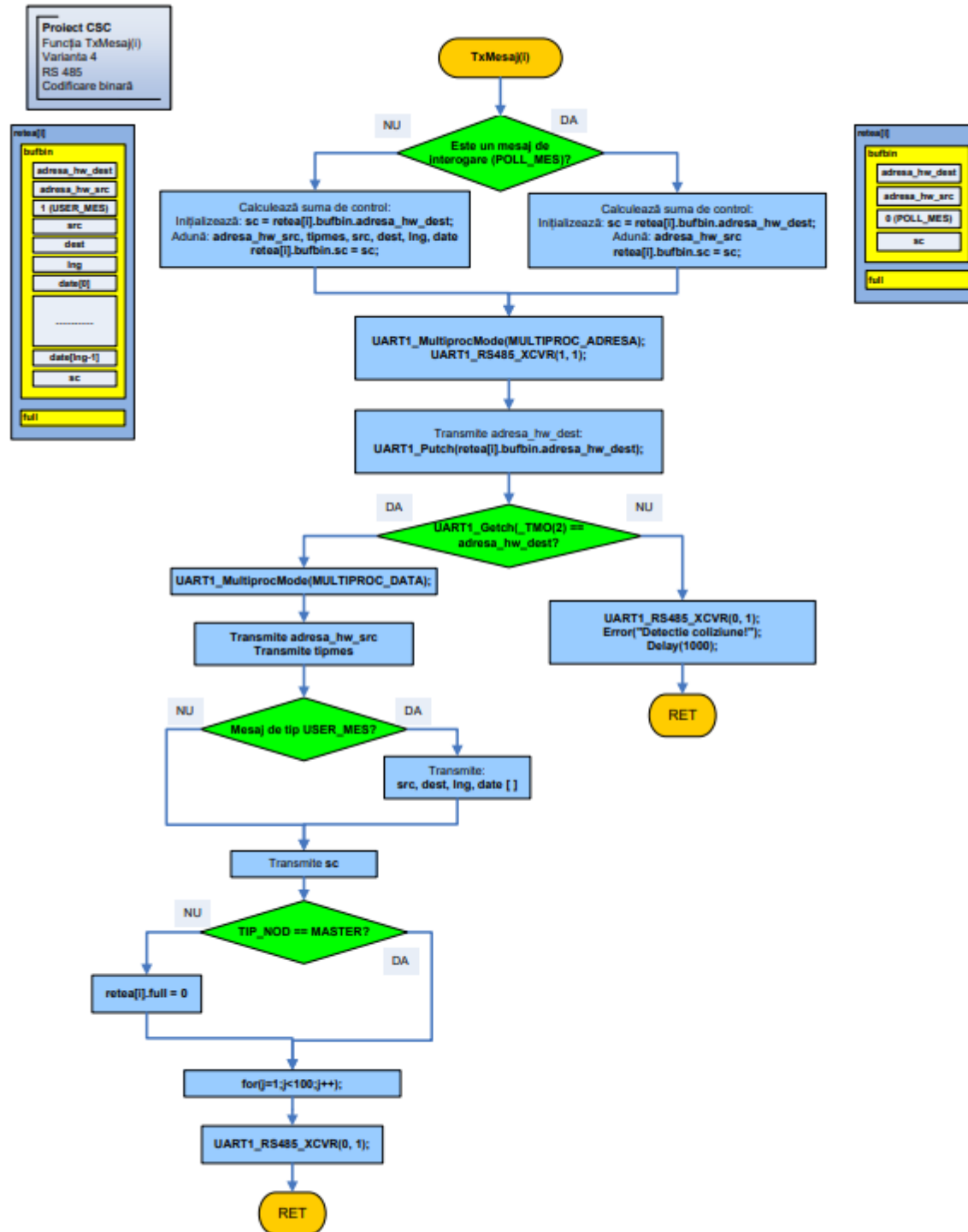
comandat pe 0, în ultimul caz rezistorul de pull-up este automat dezactivat.

- ✓ Rezistorii de pull-up pot fi toți dezactivați, indiferent de nivelul comandat pe linia PORT-OUTPUT, dacă $XBR2.7 = 1$ (bitul Weak Pull-up Disable).

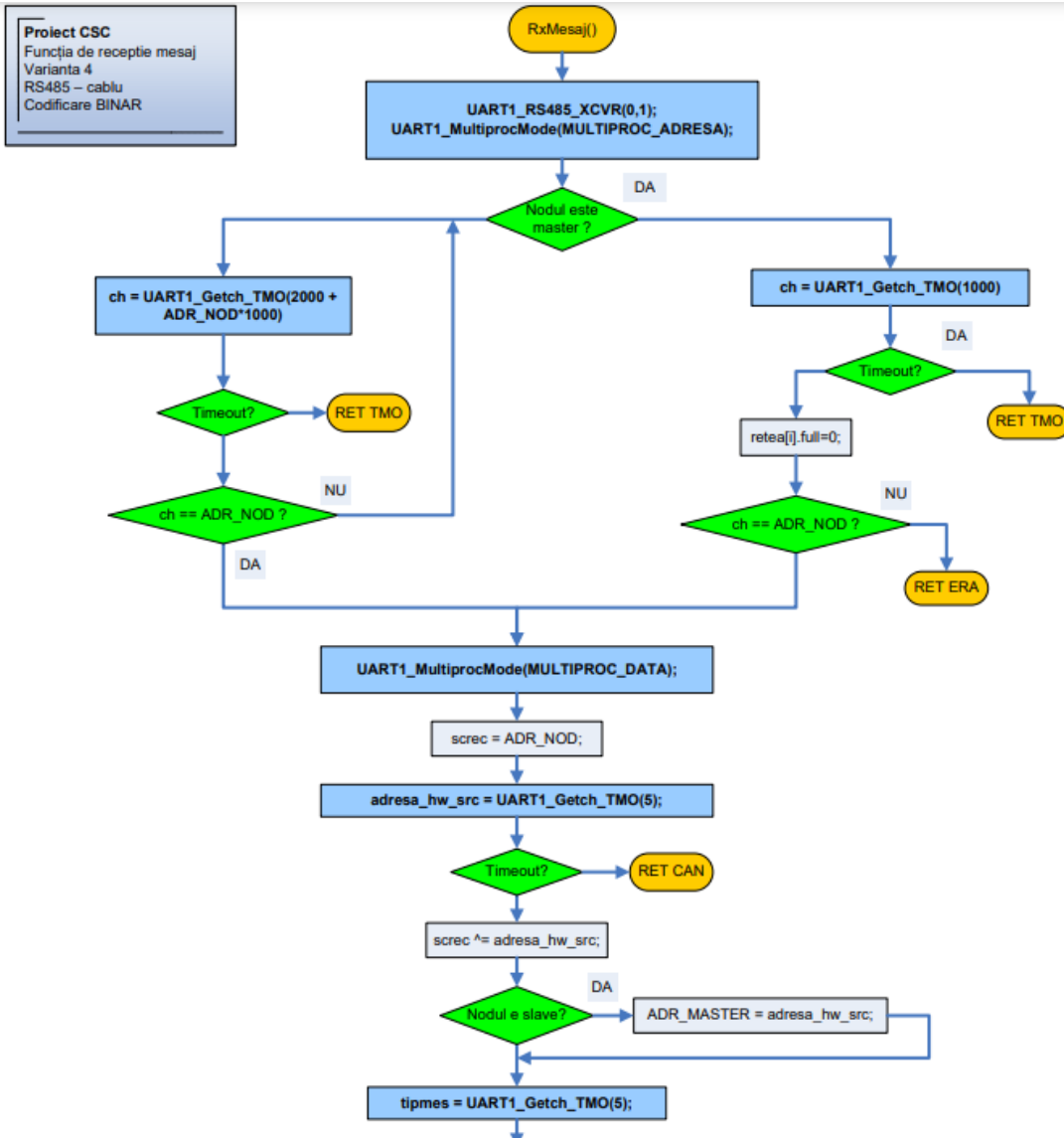
5. Programul principal

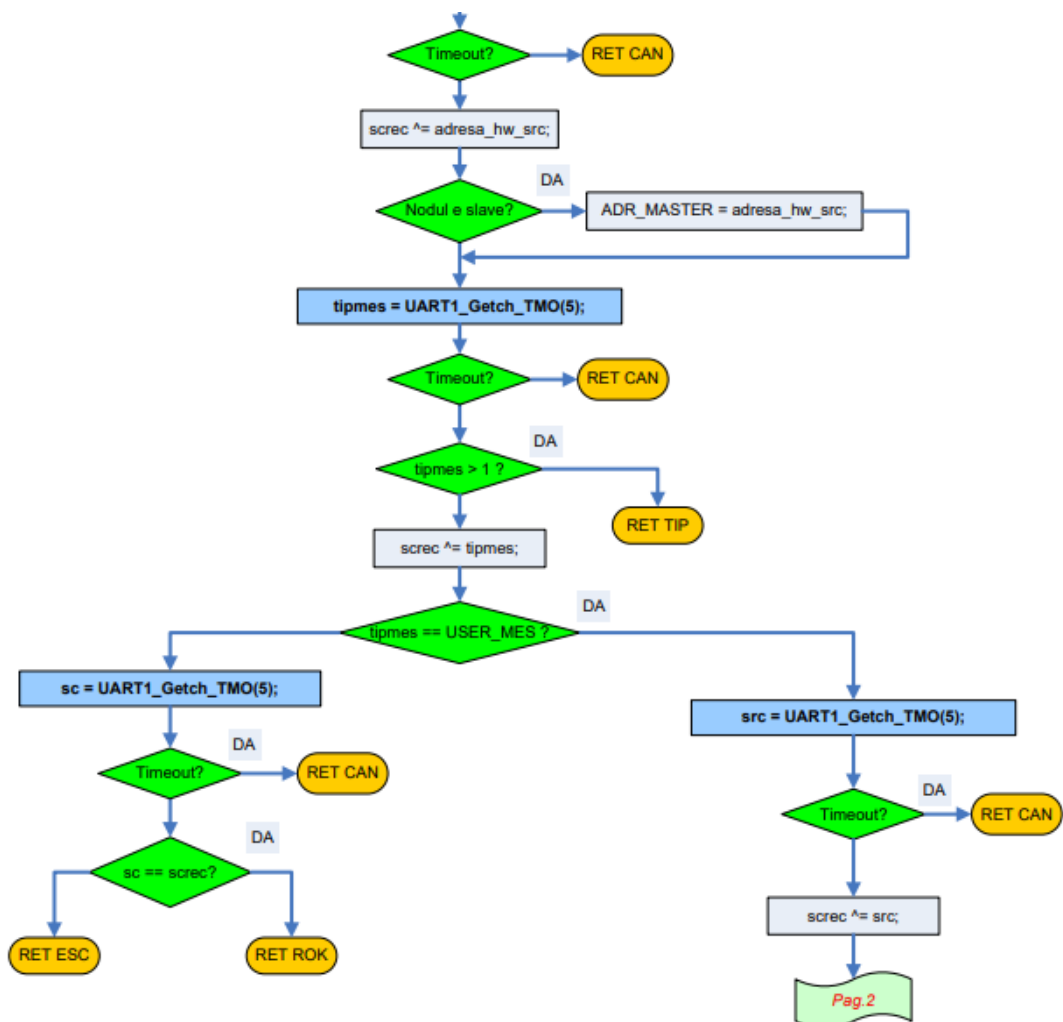


6. Pregătirea mesajelor pentru transmisie și servicii de transmisie



7. Recepția mesajelor – descriere, schema logică și implementare





8. Bibliografie

1. Suportul tehnic al microsistemului BIG8051.
2. Suportul de curs - „Comunicații în sisteme de conducere”.
3. Îndrumarul de laborator pentru disciplina „Comunicații în sisteme de conducere”.