

Sistem de Control Multidata-Multitask pe Microcontroller

Victor ABABII. Viorica SUDACEVSCHI

Technical University of Moldova

ababii@mail.utm.md, svm700@mail.ru

Abstract — În lucrare este propusă o metodă de proiectare a sistemelor de control MultiData-MultiTask în bază de dispozitive microcontroller care permite eficientizarea utilizării memoriei program. Specificul acestor sisteme de control este execuția mai multor Task-uri care operează cu mulțimi de date proprii sau comune pentru toate task-urile. În lucrare sunt prezentate: modelul matematic al sistemului de control, secvența de operații pentru proiectarea sistemului, structura logică a sistemului de control, diagrama de interacțiune a componentelor sistemului de control și algoritmul de funcționare pentru arbitru de task-uri. Pentru verificarea metodei elaborate a fost efectuată sinteza și simularea unui sistem de control MultiData-MultiTask pentru gestiunea unui robot mobil care se deplasează pe o traiectorie definită. Rezultatele obținute sunt analizate în lucrare.

Cuvinte Chee — Sistem de Control, MultiData, MultiTask, Microcontroller, Arbitru de Task-uri.

I. INTRODUCERE

Timpul este un parametru important în sistemele de control și este esențial în sistemele de timp-real. Periodicitatea execuției, timpul limită de execuție, întârzierile maxime admisibile, etc., reprezintă restricții de timp, respectarea cărora este esențială pentru îndeplinirea cu succes a sarcinilor de control. Pentru a obține o eficiență maximă este necesar de menținut permanent ocupată unitatea centrală de procesare (UCP). Orice situație de repaus duce la reducerea eficienței sistemului de control.

Pentru a spori performanțele sistemelor de control s-au dezvoltat diferite tehnici de programare, una dintre acestea fiind regimul MultiTask [1,2]. Acest regim implementează două strategii de funcționare: preemptibilă și non-preemptibilă (task-urile pot fi sau nu întrerupte de alte taskuri mai prioritare) [3,4].

Astăzi microcontrolerele reprezintă una dintre cele mai accesibile și ieftine soluții pentru implementarea sistemelor de control. Aceste dispozitive programabile prin soft sunt flexibile și asigură performanțe satisfăcătoare în ceea ce privește viteza de procesare a datelor, însă au resurse limitate de memorie EEPROM pentru stocarea programelor și de memorie SRAM pentru stocarea datelor [5,6]. Acest dezavantaj impune anumite restricții de utilizare a sistemelor de control pe microcontrolere.

În lucrare se propune o metodă de proiectare a sistemelor de control MultiData-MultiTask (MDMT) pe dispozitive microcontroller care permite micșorarea volumului de memorie program EEPROM, necesar pentru implementarea algoritmului de control. Metoda constă în definirea unui set de proceduri standarde, în care pot fi fragmentate task-urile, astfel încât pentru executarea mai multor task-uri să fie utilizat un număr optimal de proceduri comune.

II. MODELUL MATEMATIC AL SISTEMULUI DE CONTROL

Un sistem de control MultiData-MultiTask (MDMT)

este definit de mulțimea $SC = \{T, D, P\}$, unde:

$T = \{T_1, T_2, \dots, T_M\}$ reprezintă mulțimea de task-uri,

$D = \{D_1, D_2, \dots, D_M\}$ reprezintă mulțimea de date,

fiecare element al căreia aparține anumitui task

$\{D_i \in T_i, \forall i = \overline{1, M}\}$ și $P = \{P_1, P_2, \dots, P_N\}$

reprezintă mulțimea de proceduri.

Mulțimea de proceduri P este obținută din expresia

$P = \{P_{T_1} \cup P_{T_2} \cup \dots \cup P_{T_M}\}$, unde

$P_{T_i} = \{p_j, \forall j = \overline{1, M_{T_i}}\}, \forall i = \overline{1, M}$ - mulțimea de

proceduri P_j care fac parte din taskul T_i , M_{T_i} - numărul

total de proceduri în taskul T_i , M - numărul total de task-

uri, N - numărul total de proceduri ale sistemului.

Eficiența E de utilizare a procedurilor pentru sistemul MTMD este calculată în baza formulei:

$$E = \frac{\sum_{i=1}^M M_{T_i}}{N} \quad (1)$$

Secvența de operații pentru proiectarea sistemului de control MDMT este prezentată în Figura 1, unde: 1 – definirea mulțimii de task-uri executate de sistemul de control SC ; 2 – fragmentarea task-urilor în scopul identificării procedurilor comune pentru diferite task-uri; 3 – gruparea procedurilor după funcționalitate; 4 – verificarea dacă numărul de proceduri comune pentru toate task-urile T este optimal; 5 – lista de task-uri, executate de sistemul de control și a procedurilor care fac parte din taskul respectiv; 6 – lista de proceduri P , definite pentru toate task-urile.

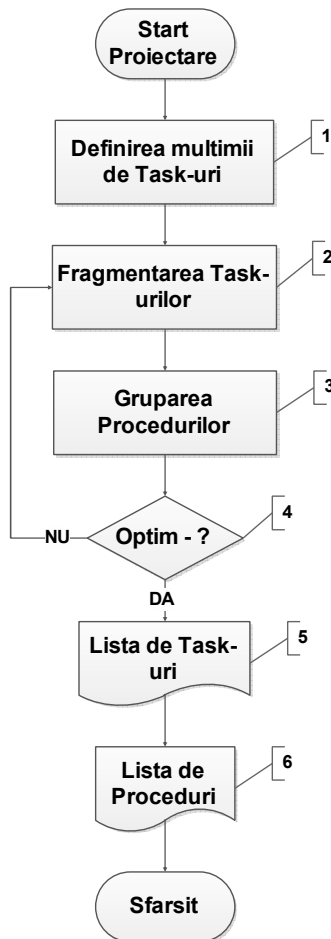


Fig. 1. Secvența de operații pentru proiectarea sistemului de control.

III. STRUCTURA LOGICĂ A SISTEMULUI DE CONTROL

Structura logică a sistemului de control MDMT este prezentată în Figura 2 și include următoarele blocuri funcționale: **Procesor** – unitatea centrală a dispozitivului microcontroler care execută secvența de instrucțiuni ale task-ului; **Proceduri** – memoria pentru stocarea procedurilor P care formează secvența de instrucțiuni ale task-urilor; **Arbitrul de task-uri** – programul manager care coordonează ordinea de execuție a procedurilor pentru îndeplinirea tuturor instrucțiunilor din Task-uri; **Data Task** – memoria pentru stocarea datelor D care include: **Data Stare** – datele de stare ale Task-ului; **Data In** – datele de intrare ale task-ului (aceste date sunt generate de instrucțiunile task-ului destinate pentru operații de intrare); **Data Out** – datele de ieșire ale task-ului (aceste date sunt generate de instrucțiunile task-ului destinate pentru operații de ieșire); **Parametrii Task-ului** – include secvența de proceduri P_{T_i} ale task-ului T_i și condițiile de intrare în task, așteptare în task și ieșire din task.

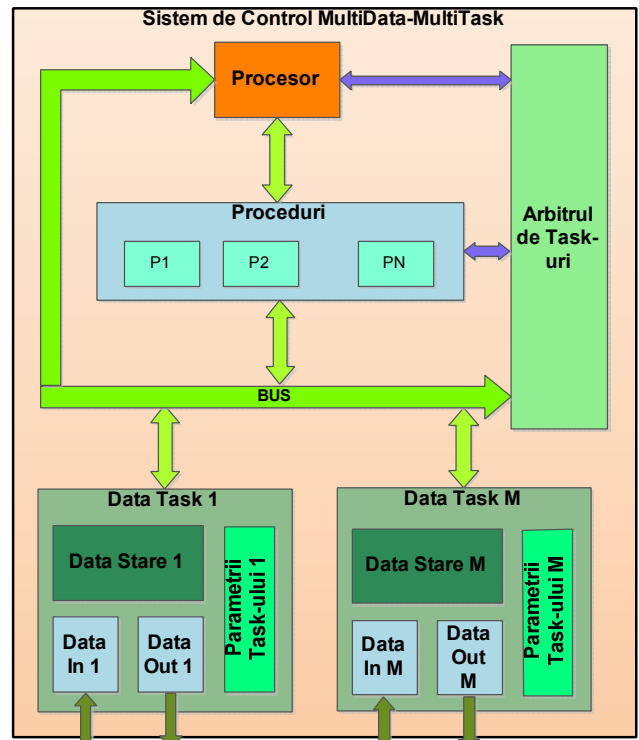


Fig. 2. Structura logică a sistemului de control.

IV. DIAGRAMA DE INTERACȚIUNE A COMPONENTELOR SISTEMULUI DE CONTROL

Diagrama de interacțiune a componentelor sistemului de control MultiData-MultiTask este prezentată în Figura 3, unde: **1. Secvența Task** – arbitrul task-ului citește parametrii din Data Task; **2, 8. Start Task** – începe execuția task-ului; **3, 9. Exec Task** – execuția instrucțiunilor din task; **4, 10. Date pentru procesare** – citirea datelor pentru task; **5, 11. Procesare Date** – procesează datele task-ului; **6, 12. Rezultatul procesării** – transmiterea rezultatului procesării datelor; **7. Sfârșitul procedurii k** – informarea Arbitrului de task-uri despre terminarea executării procedurii curente.

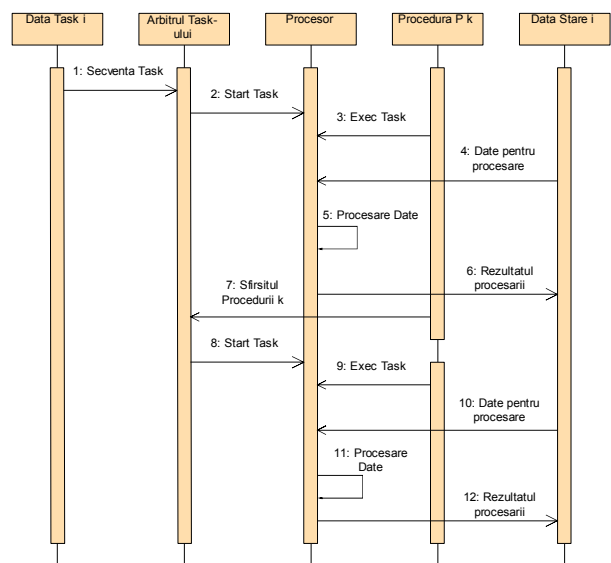


Fig. 3. Diagrama de interacțiune a componentelor sistemului de control.

V. ALGORITMUL DE FUNCȚIONARE AL ARBITRULUI

Schema bloc pentru algoritmul de funcționare al arbitrului de task-uri este prezentată în Figura 4, unde: 1 – încărcarea parametrilor task-ului T_i ce urmează a fi executat; 2 – formarea unui ciclu din M_{T_i} iterații pentru executarea procedurilor ce fac parte din task-ul T_i ; 3 – executarea procedurii p_j ; 4 – procesarea datelor D_i .

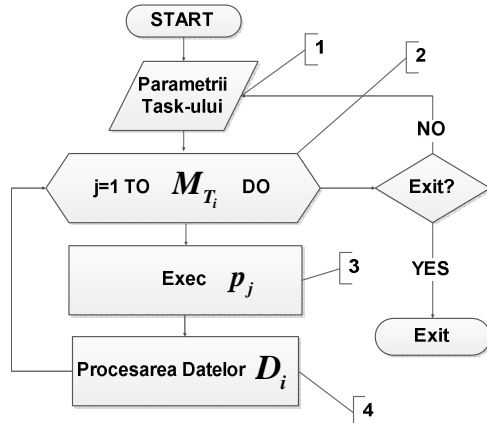


Fig. 4. Algoritmul de funcționare al Arbitrului.

VI. EXEMPLU DE SINTEZĂ A UNUI SISTEM DE CONTROL

În calitate de exemplu de sinteză a unui sistem de control MTMD se propune gestiunea unui robot mobil care se deplasează pe o traiectorie definită [7]. În rezultatul analizei modelului matematic de descriere a traiectoriei de deplasare a robotului mobil au fost selectate trei task-uri $T = \{T_1, T_2, T_3\}$. În rezultatul fragmentării task-urilor a fost obținută următoarea distribuție a procedurilor:

$$T_1 = \{P_1, P_3, P_6\},$$

$$T_2 = \{P_2, P_3, P_4, P_5\},$$

$$T_3 = \{P_1, P_2, P_5, P_6\}.$$

Secvența executării procedurilor în task-urile sistemului de control pentru gestiunea robotului mobil este prezentată în Figura 5, unde avem trei taskuri și șase proceduri, obținute în rezultatul fragmentării.

În baza formulei (1) poate fi calculată eficiența sistemului de control MTMD care constituie:

$$E = \frac{11}{6} \approx 1,8.$$

VII. CONCLUZII

Utilizarea eficientă a memoriei program și de date în sistemele de control realizate pe microcontrolere este foarte importantă, având în vedere resursele limitate de memorie în aceste dispozitive. Metoda de proiectare a sistemelor de control MultiData-MultiTask, propusă în această lucrare

permite o utilizare mai productivă a memoriei program din dispozitivele microcontroler. Metoda se bazează pe faptul că task-urile, ce fac parte din algoritmul de control, sunt divizate în mai multe proceduri standard. Executarea algoritmului de control constă în îndeplinirea mai multor task-uri, fiecare dintre care procesează o mulțime de date proprii sau comune pentru toate task-urile, utilizând aceleași proceduri standard.

În perspectivă se planifică cercetarea și identificarea punctelor critice de concurență și optimizarea memoriei de date.

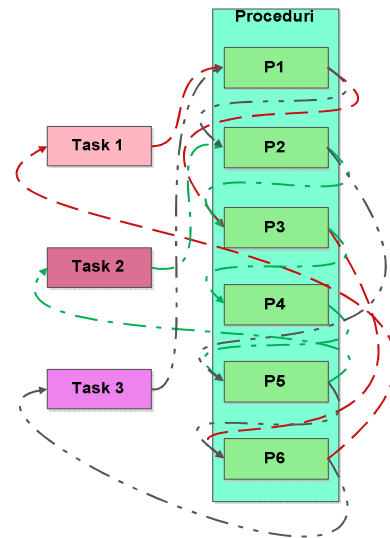


Fig. 5. Secvența de execuție a procedurilor în task-urile sistemului de control [7].

MENȚIUNI

Rezultatele obținute în această lucrare fac parte din activitățile efectuate în cadrul proiectului bilateral Moldova – Ucraina 14.820.18.02.03/u „Rețele de senzori distribuite cu noduri de calcul reconfigurabile”.

REFERINȚE

- [1] Martin P. Bates. Programming 8-bit PIC Microcontrollers in C: with Interactive Hardware Simulation, Newnes, ISBN: 978-0-7506-8960-1, 2008, 304p.
- [2] Keith E. Curtis. Embedded Multitasking: with small Microcontrollers, Newnes, ISBN: 0-7506-7918-2, 2006, 417p.
- [3] <http://www.piclist.com/techref/microchip/> (Accesat 11.08.2014).
- [4] D. Kalinsky, Basic Concepts of Real-Time Operating Systems, Linux Devices, Nov. 2003; (Accesibil pe http://www.jmargolin.com/uavsjm_rpv2_npl_16.pdf).
- [5] <http://www.atmel.com> (Accesat 15.08.2014).
- [6] <http://www.microchip.com> (Accesat 15.08.2014).
- [7] А. А. Кабанов, Оптимальное управление траекторным движением мобильного робота, XII Всероссийское совещание по проблемам управления, ВСПУ-2014, Москва 16-19 июня 2014 г. стр. 3760-3770.