

ALGORITMI DE COMUNICARE ALE UNITATILOR DE COMANDA ALE MICROSATELITULUI

Alexei Cârțâca, Eugeniu Suman, Nicolae Secrieru, Roman Nicu
Universitatea Tehnică a Moldovei

cirtica@rambler.ru, jenea_ne@yahoo.com, nsecrieru@gmail.com, roman@rgg.md

Abstract. *The internal communication of a microsatellite represents the possibility of transmitting and receiving of commands and data by all the subsystems. The performances of the microsatellite's subsystems differ according to their function, that is why the communication between them should be flexible and reliable at the same time.*

Cuvinte-cheie: *calculator de bord, protocol, comunicare, redundanță, fiabilitate sporita.*

I. Introducere

Structura decentralizata a calculatorului de bord, bazata pe o retea locala de microcontrolere, include un microcontroler de baza si o serie de microcontrolere cu module de interacțiune cu unitățile periferice pentru controlul cu toate sistemele de bord ale aparatului cosmic [3, 7, 12]. Unitatea de comandă centrală interacționează cu celelalte unități de comandă periferice printr-un canal comun. Unitățile sistemului vor interacționa pe canal de o lungime scurtă de 5-10 cm, însă într-un mediu cu perturbații de un nivel înalt. Ca urmare, perturbațiile din acest mediu pot provoca erori în mesajul transmis prin canalul de comunicare. Protocolul proiectat trebuie să asigure transmiterea corectă a datelor între unitățile sistemului distribuit de comandă.

II. Estimarea parametrilor comunicației interne a calculatorului de bord

Inițial să estimăm proprietățile canalului de legătură. Din punctul de vedere a teoriei informației canalul de transmisiune se caracterizează prin setul x de semnale la intrare, setul y de semnale la ieșire și setul de probabilități condiționate $p(y/x)$ de recepție a semnalului având la intrare semnalul x . Aceste probabilități descriu proprietățile statice ale „zgomotului”, a perturbațiilor, care distorsionează semnalele în procesul de transmisiune. În cazul, când $p(y/x) = 1$ pentru $y = x$ și $p(y/x) = 0$ pentru $y \neq x$, avem un canal ideal fără „zgomot”. În mod similar: pentru un canal real avem: $p(y/x) = 1$ pentru $y = x$ și $p(y/x) = 0$ pentru $y \neq x$. Având aceste valori ale probabilităților, se va pute determina capacitatea canalului – proprietatea de a transmite o cantitate de informație într-o unitate de timp, care se determină cantitatea de informație maximă în semnalele de la ieșire relativ de semnalele de intrare într-o unitate de timp. Se va determina în modul următor [5]. Considerăm că semnalul de intrare x care primește valorile x cu probabilitate $p(x)$. Conform teoriei probabilității se va calcula probabilitatea $q(y)$ că semnalul h va avea la ieșire valoarea y :

$$q(y) = \sum p(x) \cdot p(y|x) \quad (1)$$

la fel și probabilitatea $p(x, y)$ evenimentelor concomitente $x = x, h = y$:

$$p(x, y) = p(x) p(y|x). \quad (2)$$

După acestea ecuații se poate determina cantitatea de informație și valoarea medie (în alfabet binar), considerînd că :

$$I(h, x) = I(xh) \quad (3)$$

$$R = \lim_{T \rightarrow \infty} \frac{1}{T} \cdot I(h, x) \quad (4)$$

unde T – durata semnalului x .

Limita de sus a valorii R luată pentru toate posibilele semnale la intrare se numește capacitatea canalului, și se determină din următoarea relație:

$$C = \lim_{T \rightarrow \infty} \frac{\log_2 N(T)}{T} = \lim_{nt \rightarrow \infty} \frac{n}{nt} \quad (5)$$

unde $N(T)$ – numărul semnalelor cu durata $T = nt$, dacă se prezintă în cod binar $\{0,1\}$.

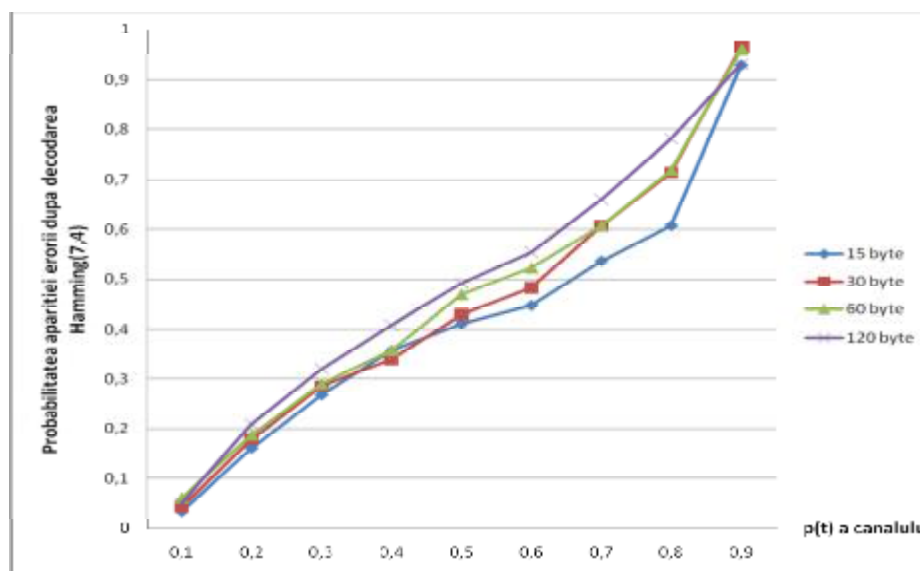


Figura 1. Dependența probabilității apariției erorii de lungimea mesajului

Ținând cont de condițiile din problema formulată: unitatea de comandă trebuie să fie capabilă să recepționeze o cantitate de informație pînă la 256 baiți de la 1-:-255. Tot odată, trebuie să fie posibilitatea de a modifica viteza de transmitere. Analizînd tipul de canale fizice utilizate pentru microcontroler, cele mai performante sunt RS-485/232, care pot să asigure la astfel de distanță pînă la 115 kbod. Luînd în considerație ratele de viteze de comunicare prin porturile seriale, au fost selectate ratele de transmitere de 9600, 19200 și 38400 bod.

Următoarea problemă în proiectarea protocolului de comunicare este determinarea lungimei mesajelor, ținînd cont de volumul de informație și nivelul de perturbații. Această problemă are o soluție analitică. Dacă se va determina setul de probabilități condiționate $p(y/x)$, atunci putem aprecia această lungime. Dar această soluție are două laturi negative: este complicat de apreciat aceste probabilități, trebuie de efectuat un număr mare de experimente, este teoretică – nu ține cont de dinamica proceselor de transmisiune.

Problema dată o soluționăm prin simulare, făcînd unele aprecieri a probabilității de perturbare a semnalelor. Soluționarea se efectuează după următoarea schemă: se simulează transmisiunea datelor pe o perioadă anumită, de exemplu $t = 60$:-100 sec. Datele se grupează în mesaje de anumită lungime. Concomitent cu transmisiunea datelor se simulează perturbațiile: dacă la transmiterea unui bit perturbația a depășit pragul-limită, bitul dat se notifică. Transmisiunea poate fi efectuată fără codificare sau cu codificare detectoare de erori, de exemplu, codul Hamming, care poate detecta 2 erori în cadrul unui byte. Trebuie de ținut cont, că codificarea majorează lungimea mesajului de 2 ori. Dacă în cadrul unui mesaj a apărut erori de perturbare, care nu se pot corecta, atunci mesajul se retransmite.

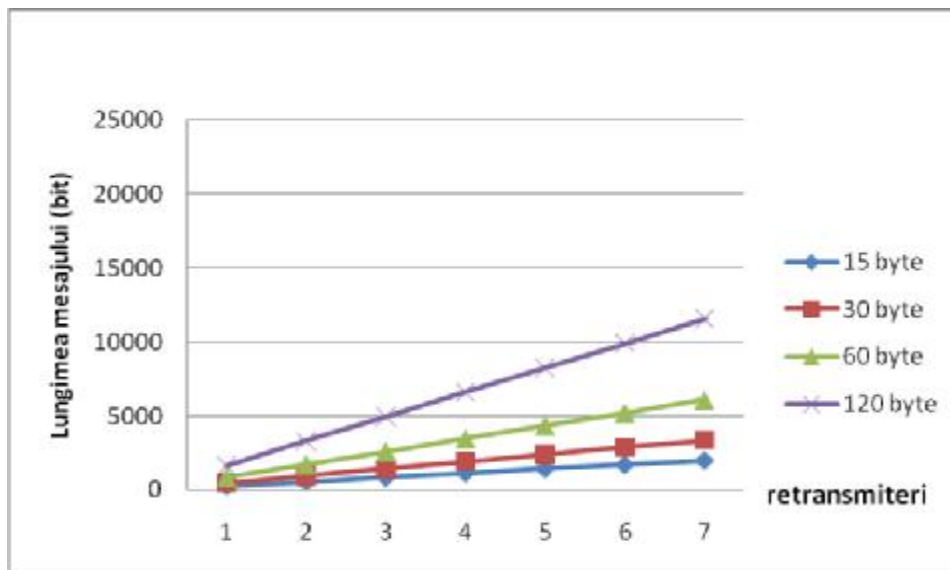


Figura. 2. Creșterea lungimii mesajului la retransmiterea mesajului necodificat

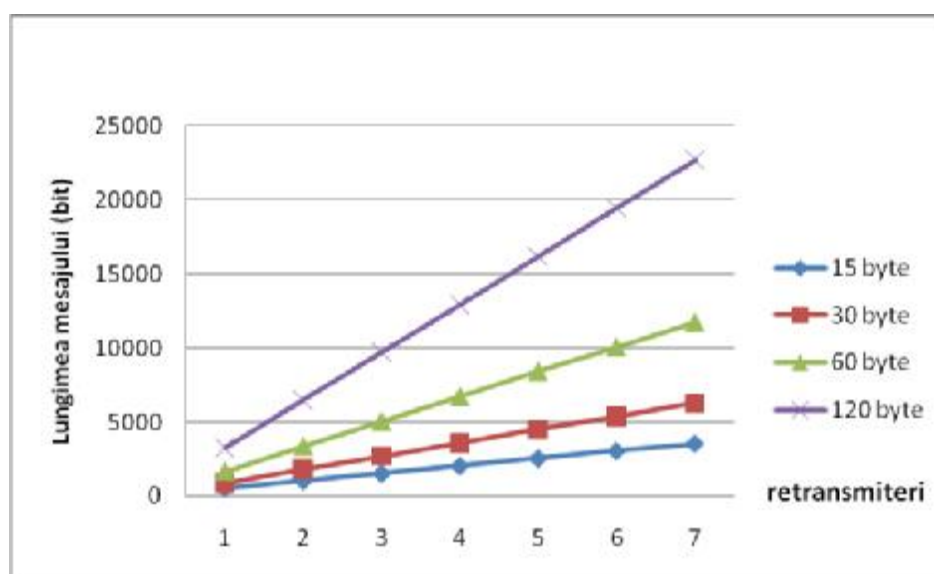


Figura 3. Creșterea lungimii mesajului la retransmiterea mesajului codificat Hamming(7,4)

Presupunem că transmisia cadrelor este independentă, iar ρ va indica probabilitatea că un oarecare cadru va fi pierdut sau va fi primit cu erori care pot fi detectate.

Vom considera timpul necesar sistemului A pentru un singur cadru egal cu T , t_f – timpul necesar pentru transmiterea cadrului, t_{out} – intervalul de așteptare a răspunsului (*timeout*). Considerând transmisia continuă, fără pauze între transmiterea a două cadre succesive, timpul între două transmisiuni va fi egal cu t_f . La expirarea intervalului de *timeout* fixat are loc recepționarea confirmării sau retransmiterea cadrului. Considerăm $t_{out} = \beta \cdot t_f$, unde raportăm intervalul de *timeout* la timpul de transmitere a unui cadru prin intermediul unui parametru pozitiv β . Acum considerăm că sunt necesare K retransmisii pentru cadrul pe care îl transmitem curent, adică primele K transmiteri ale cadrului au eșuat (pachete pierdute sau erori în cadru), iar a $K+1$ încercare a fost cu succes (lungimea

mesajului codificat și necodificat sunt prezentate în figurile 3 și 4). Atunci, timpul total necesar pentru ca terminalul A să termine cu succes transmiterea unui cadru este:

$$T = t_f + K(t_f + t_{out}) = t_f + (1 + b)Kt_f. \quad (6)$$

Se observă, că numărul de retransmiteri K este o distribuție geometrică:

$$P\{K = k\} = p^k(1 - p) \quad (7)$$

unde $(k = 0, 1, \dots)$

pentru transmiterea independentă a cadrelor. Urmează că timpul mediu utilizat de sistemul A pentru un singur cadru este

$$\bar{T} = E(T) = t_f + t_f(1 + b)E(K) = t_f + t_f(1 + b)\sum_{k=0}^{\infty} kp^k(1 - p) = t_f + t_f(1 + b)p(1 - p)\sum_{k=0}^{\infty} kp^{k-1} \quad (8)$$

Probabilitatea detectării erorilor depinde într-o măsură și de lungimea mesajului. Să analizăm cazul transmiterii mesajului cu diferite lungimi printr-un canal cu distorsiuni. Vom analiza cazul cu lungimile 15, 30, 60 și 120 bytes cu probabilitatea apariției erorilor de la 0,1 pînă la 0,9 cu codarea Hamming(7,4). Rezultatele simulării sunt prezentate în tabelele 1, 2 și 3, grafic rezultatele sunt prezentate în Figura 3.

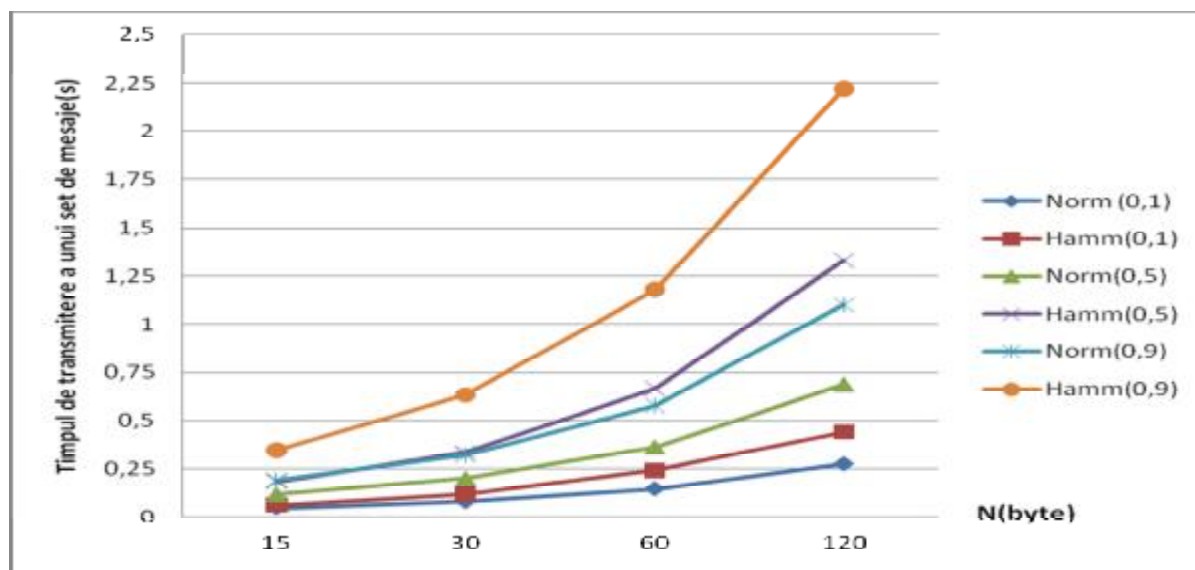


Figura 4. Dependența timpului mediu de transmiterea mesajului de lungimea mesajului și nivelul perturbărilor.

Este evident că se analizează timpul total de transmitere pentru viteza cea mai mică, în cazul de față este de 9600 baud/s.

Tabelul 1: Rezultatele simulării pentru probabilitatea $p=0,1$.

Lung. Timp	15 byte	30 byte	60 byte	120 byte
Norm (s)	0,047667	0,080167	0,145167	0,275167
Hamm (s)	0,063903	0,118464	0,240195	0,441439

Tabelul 2: Rezultatele simulării pentru probabilitatea $p=0,5$.

Lung. Timp	15 byte	30 byte	60 byte	120 byte
Norm (s)	0,119167	0,200417	0,362917	0,687917
Hamm (s)	0,182953	0,333721	0,666049	1,32953

Tabelul 3: Rezultatele simulării pentru probabilitatea $p=0,9$.

Lung. Timp	15 byte	30 byte	60 byte	120 byte
Norm (s)	0,190667	0,320667	0,580667	1,100667
Hamm (s)	0,347063	0,634048	1,180678	2,221324

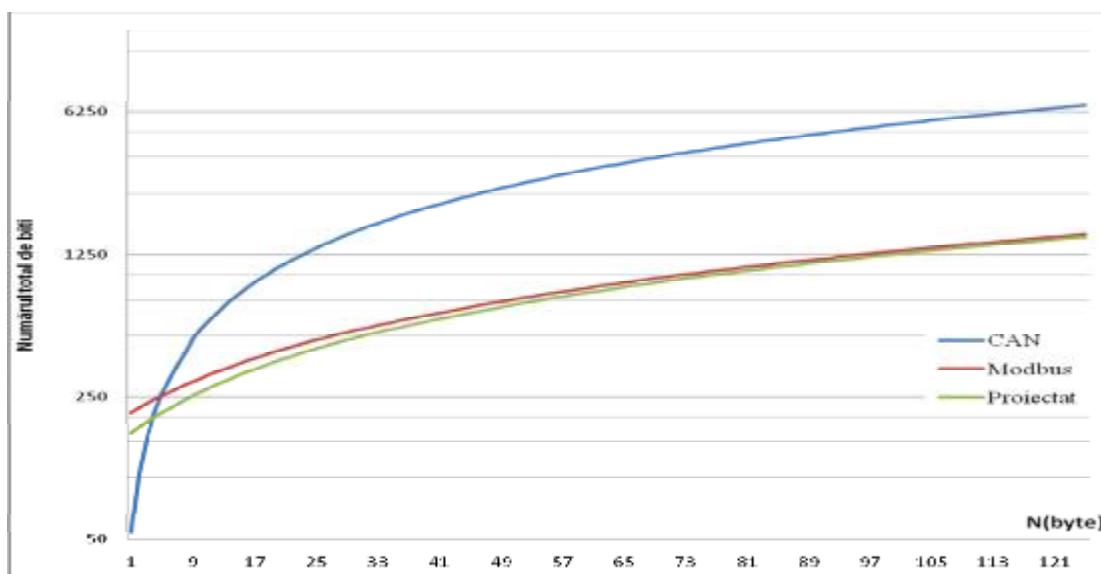


Figura 5. Graficul dependenței lungimii reale a mesajului de lungimea de date utile cu aplicarea diverselor protocoale de comunicare: CAN, Modbus RTU și protocolul propus.

Pentru estimarea timpului de transmitere a mesajului față de protocoale existente, vom efectua compararea timpului de transmitere a N bytes de date prin intermediul protocolului elaborat, CAN și Modbus RTU (fig. 5). E necesar de menționat următoarele.

CAN - Mesajul standard a protocolului CAN conține un bit de start SOF, segment de arbitraj (12 bit), segment de control (6 bit), cimp de date (max 8 bytes), cimpul sumei de control CRC (15 bit), cimp de confirmare ACK (2 bit) și cimp de sfârșit a mesajului EOF (7 bit). Distanța minimă între mesaje este de 3 biți. Ca rezultat, pentru transmiterea de N bytes de date sunt necesare N mesaje și N-1 distanțe minime între mesaje: $(43+64)N+3(N-1)=110N-3$ (bit).

Modbus RTU - Mesajul protocolului Modbus începe și se termină cu interval de tăcere de $3.5 \cdot 11$ (biti). Mesajul este alcătuit din cimp de adresa (1 byte), cimpul codului comenzii (1 byte), cimp de date (pina la 252 bytes), cimpul sumei de control CRC (2 bytes). la transmiterea a $8 \cdot N$ bytes date, sunt necesari $(7+4+N)11=11N+121$ (bit).

In protocolul elaborat este posibilă transmiterea a unui șir de 255 bytes date într-un mesaj. Deoarece la transmiterea prin RS-485 la fiecare byte se adaugă 3 biți, 1bit - start și 2biți - stop (1 bit - stop, în cazul cînd se utilizează bit de paritate), se transmit 11 biți. Deci, la transmiterea a $8 \cdot N$ bytes date, sunt necesari $(7+N)11=11N+77$ (bit).

Protocolul elaborat permite micșorarea numărului de biți utilizați pentru transmiterea datelor.

în mesaje scurte[12], specifice satelitului: de exemplu, pînă la 128 bytes date – 80%(CAN), 2,5%(Modbus), iar pînă la 256 bytes date – 78%(CAN), 2%(Modbus).

Efectuînd analiza rezultatelor simulării și structurii mesajului putem face concluzia că lungimea maximă recomandată a mesajelor dintre unitățile de comandă nu trebuie să depășească intervalul de 60 -:- 80 byte.

III. Concluzii

Efectuînd o analiză a experienței altor dezvoltatori de microsateliți, la fel și analiza sarcinilor de calcul impuse de misiunea spațială, permite de face concluzia, că pentru astfel de sisteme este satisfăcător un model de comunicație cu trei nivele. Protocolul elaborat permite micșorarea numărului de biti utilizați pentru transmiterea datelor în mesaje scurte, specifice satelitului: de exemplu, pînă la 128 bytes date – 15%(CAN), 2%(Modbus), iar pînă la 256 bytes date – 17%(CAN), 1%(Modbus), deci utilizarea protocolului propus devine efectivă la transmiterea pachetului de date cu lungimea $l = 40\text{--}128$ byte. Fiabilitatea transmiterii datelor sporește datorită utilizării codării Hamming și a transmiterii repetate de date.

IV. Referințe

1. Alminade, L., Bisgaard, M., Binther, D., Viscor, T., and Sstergaard, K. - In: Robustness of Radio Link Between AAU-Cubesat and Ground Station.
2. Aoki, Y., Barza, R., Zeiger, F., Herbst, B, and Klaus, S. *The CubeSat Project at the University of Wurzburg: The Mission and System Design.*- In: <http://www7.informatik.uni-wuerzburg.de/cubesat/>
3. B. Herbst, F. Zeiger, M. Schmidt, K. Schilling, "UWE-1: A Picosatellite to Test Communication Protocols," Proceedings of the 56th IAC 2005, Fukuoda, Japan, Oct. 17-21, 2005, IAC-05-B5.6.A.11
4. K. Schilling, M. Schmidt, R. Barza, "In orbit experiences from the picosatellite UWE-1," Proceedings of the 4S Symposium: 'Small Satellite Systems and Services,' Chia Laguna Sardinia, Italy, Sept. 25-29, 2006, ESA SP-618
5. B. Palmintier, et. al., "Distributed Computing on Emerald: A modular approach for Robust Distributed Space Systems"- In Proceedings of the 2000 IEEE Aerospace Conference, Big Sky, MO, March 2000.
6. Briess, K., Montenegro, S., Bärwald, W., Halle, W., Kayal, H., Lorenz, E., Skrbek, W., Studemund, H., Terzibaschian, T., Walter, I.: *Demonstration of Small Satellite Technologies by the BIRD Mission* – In: 16th Annual AIAA/USU Conference on Small Satellites, Utah, USA 2002
7. Burton I. Edelson, Joseph N. Pelton. *Satellite communications systems and technology* - Europe, Japan, Russia. Noyes Data Cooperation, 1995
8. Huera, Derek. *Development of a Highly Integrated Communication System for use in Low Power Space Applications.*- In: Cal Poly Masters Thesis. April 2006.
9. Kitts, C., et al. Flight Results from the GeneSat-1 Biological Microsatellite Mission.- In Proceedings of the 21st AIAA/USU Conference on Small Satellites. SSC07-XI-1. August 2007.
10. Mas, I., and Kitts, C. *A Flight-Proven 2.4GHz ISM Band COTS Communications System for Small Satellites.*- In: Proceedings of the 21st AIAA/USU Conference on Small Satellites. SSC07-XI-11. August 2007.
11. Noe, Chris. *Design and Implementation of the Communications Subsystem for the Cal Poly CP2 Cubesat Project.* Cal Poly Senior Project. June 2004.
12. Cârțâca Alexei, Suman Eugeniu, *Protocolul și algoritmiile de comunicare ale unitatilor de comanda ale microsateitului.* - In: 3rd International Conference „ICTEI-2010”, Chisinau, 2010.