

Modul Configurabil de Recepție a Datelor Telemetriche Satelitare

Ilco V., Levineț N.

Centrul Tehnologii Spațiale
Universitatea Tehnică a Moldovei
Chișinău, Republica Moldova
valentin.ilco@gmail.com, levinet.nicolai@gmail.com

Abstract — In this paper is represented architecture of a reconfigurable module for reception of satellite telemetry data based on the software architecture. Therefore, for telemetry receive by AX.25 protocol, Doppler correction, demodulation and data decapsulation blocks are creating using programming languages. The transition from the hardware to the software part, offers the possibility to amend telemetry station operation depending on the input data for each block. This module can be configured to change the filter parameters, change the principle of operation of the demodulator or reconfiguration of data decapsulation method depending on the satellite communication protocol. The use of software defined radios would improve the flexibility on the communication process, and allow support for new waveforms to be added without hardware upgrades or intervention of an operator at every ground station.

Keywords — Software Defined Radio, GNU Radio, USRP, AFSK, Python, GRC, AX.25, Doppler Shift.

I. INTRODUCERE

Recepția datelor telemetriche satelitare se realizează printr-o arhitectură radio. Asigurarea funcționalității unei arhitecturi radio tradiționale, în cadrul unei comunicații satelit, se bazează în primul rând pe componentele hardware, cu configurabilitate minimă prin software. Convențional, acestea constau din: modulator, codare, amplificatoare, filtre, mixere, etc., dedicate unui anumit mod de transmisie. Arhitectura radio tradițională este prezentată mai jos:

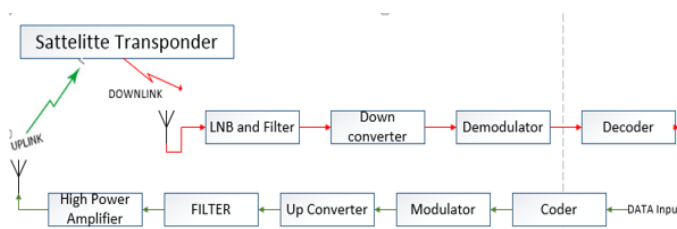


Fig. 1. Arhitectura radio tradițională de comunicare cu microsateți.

Partea software este dedicată controlului interfețelor cu rețeaua de comunicație. Ținând cont că în acest model predomină componentele hardware, un eventual upgrade al sistemului ar însemna practic schimbarea totală a modelului și reproiectarea sistemului. SDR sau Software Defined Radio rezolvă această problemă.

SDR prezintă o arhitectură flexibilă, în care funcțiile de nivel fizic sunt definite, parțial sau în totalitate, prin software.

Drept urmare, echipamentul SDR suportă o varietate de servicii și standarde, fiind totodată ușor de reconfigurat. Arhitectura flexibilă a SDR-ului se bazează pe circuite universale ce pot fi configurate sau programate software.

O stație telemetrică bazată pe un SDR, cum ar fi Universal Serial Radio Peripheral (USRP), poate fi programată pentru transmiterea și recepționarea oricărui semnal ce se găsește în banda de frecvență corespunzătoare.

USRP-ul este de obicei controlat software prin mediul GNU Radio. În cadrul acestui mediu pot fi create aplicații folosind instrumentariul grafic GNU Radio Companion (GRC). Acesta utilizează un sistem de blocuri funcționale pentru procesarea semnalelor ce alcătuiesc un grafic de flux. GRC permite proiectarea unui sistem radio prin interfața "drag and drop". După finisarea proiectării, GRC, convertează fluxul grafic într-un fișier Python, conținând descrierea corespunzătoare a sistemului. GNU Radio permite totodată crearea unor noi blocuri funcționale, care ar implementa noi operații de procesare a semnalelor. [1]

II. STAȚIE TELEMETRICĂ BAZATĂ PE SDR

Arhitectura convențională a unei stații telemetriche bazate pe SDR, este prezentată mai jos:

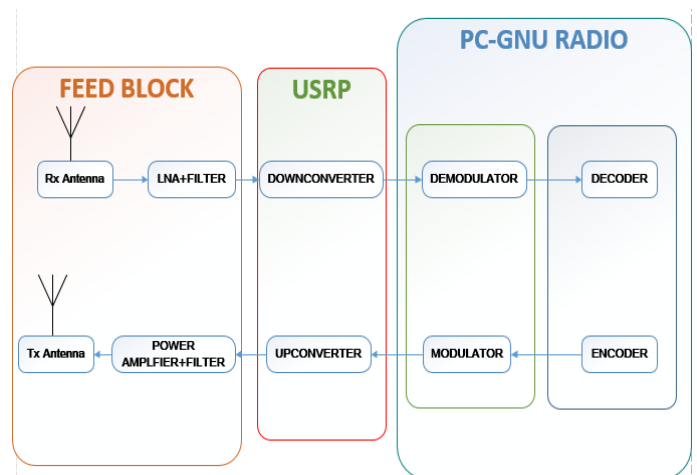


Fig. 2. Arhitectura Software Defined Radio.

În arhitectura de mai sus se diferențiază subsistemul analogic (feed block) și subsistemul digital (USRP și PC-GNU Radio). Subsistemul analogic servește ca interfață pentru mediul radiofrecvență. USRP-ul, bazat pe FPGA (Field

Programmable Gate Array), din punct de vedere hardware, prezintă o concepție de construcție modulară, fiind format dintr-o placă de bază (motherboard) la care se pot atașa mai multe tipuri de interfețe radio (cunoscute ca module daughterboard). USRP-ul este conectat la un PC printr-o interfață Ethernet Gigabit sau USB. Aici GNU radio operează cu procesarea ulterioară a semnalului, prin blocurile funcționale create. Conceptual, blocurile procesează fluxuri infinite de date ce le tranzitează de la porturile de intrare la porturile de ieșire.

Printre atributele ce caracterizează un bloc de se numără tipul de date ce îl parcurge, precum și numărul de porturi de intrare și de ieșire. În pachetul GNU Radio sunt incluse aproximativ 100 de blocuri de procesare digitală a semnalelor. Pe lângă blocurile implicate în mediul GNU Radio se pot crea blocuri noi, personalizate.

Schema convențională în mediul GNU Radio pentru recepția informației telemetrice se prezintă mai jos:

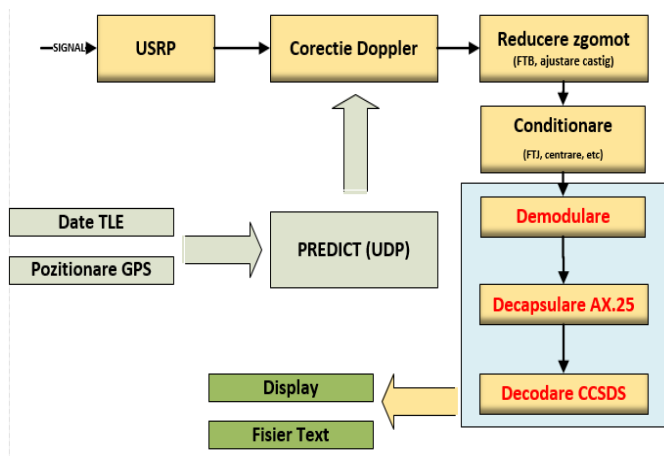


Fig. 3. Modelul de recepție a informației telemetrice în GNU Radio.

Blocul USRP prezintă o modalitate interactivă de interconectare cu modelul hardware și de configurare a interfeței de recepție a SDR-ului, totodată asigurând configurarea parametrilor esențiali cum sunt: sursa semnalului de referință, numărul de canale de achiziție (dat de tipurile de daughterboard-uri instalate), frecvența centrală a fiecărui canal, rata de eșantionare a semnalului în banda de bază (ce determină și banda semnalului achiziționat). Blocul de corecție Doppler este folosit pentru ajustarea variației frecvenței la mișcarea satelitului, fiind direct influențat de datele despre orbita satelitului. Algoritmul de corecție a semnalului se bazează pe mai multe blocuri de procesare standard (filtre, bucle). Blocurile delimitate nu se găsesc în bibliotecile standard ale GNU Radio și trebuie proiectate. În tabelul următor se prezintă blocurile de procesare digitală ce urmează a fi proiectate și implementate.

Tipul cadrului de mai sus este UI (Unnumbered Information), ceea ce permite conexiunea multiplă cu stațiile terestre. Câmpurile *Flag* delimitează fiecare secvență, câmpul *Adress* conține sursa și destinația fiecărei secvențe specifice, câmpul *PID* identifică tipul protocolului (dacă există) de pe

nivelul superior al lui AX.25, *FCS* servește pentru verificarea integrității secvenței, iar informația utilă se găsește în câmpul *Info*.

TABLE I. BLOCURILE DE PROCESARE DIGITALĂ A SEMNALELOR

Blocul PDS	Descrierea
AX.25 Demodulator	<i>Blocul extrage conținutul din semnalul modulat</i>
AX.25 Decoder	<i>Blocul extrage informația utilă din frame-ul AX.25, verifică integritatea acesteia și o redresează spre decoderul CCSDS</i>
CCSDS Decoder	<i>Blocul extrage informația telemetrică și o salvează într-un fișier sau informația este afișată într-o interfață grafică.</i>

Structura cadrului pentru protocolul AX.25 la recepționarea telemetriei este următoarea:

Flag	Address	Control	PID	Info	FCS	Flag
11110000 (0xF0)	48 bits	8 bits	8 bits	N*8 bits	16 bits	11110000 (0xF0)

Fig. 4. Structura cadrului protocolului AX.25.

III. IMPLEMENTAREA

După scrierea codului în C++ pentru crearea blocurilor, în comun cu limbajul Python pentru organizarea și interconectarea lor s-au creat următoarele blocuri de procesare digitală pentru GNU Radio:

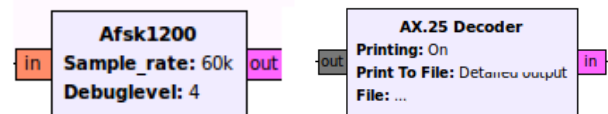


Fig. 5. Blocurile de procesare digitală create.

În figura 6 este prezentată implementarea blocurilor în modulul de recepție al informației telemetrice.

Blocul USRP Source prezintă interconectarea cu hardware-ul Ettus B200, cu parametrii definiți. Blocul *Low Pass Filter* reprezintă un filtru trece jos. *Quadrature Modulation* este folosit pentru demodularea FSK (Frequency Shift Keying), în cazul în care la transmisiune se va folosi AFSK (Audio Frequency Shift Keying) vom apela la blocul de procesare creat mai sus. Următoarele trei blocuri din schema de mai jos sunt folosite pentru analiza formei semnalului și formarea nivelurilor logice. Blocul de decodare creat permite obținerea informației telemetrice după structura de cadru a protocolului AX.25 prezentată mai sus. Modulul File Sink face posibilă stocarea datelor în timp real pe un mediu de stocare local al stației de lucru sau într-un spațiu partajat în rețea, la care are acces stația de lucru. Scrierea se face în format binar (după standardul float 32), iar datele salvate pot fi procesate ulterior fie în GNU Radio folosind un bloc de citire (File Source), fie în alte medii de programare.

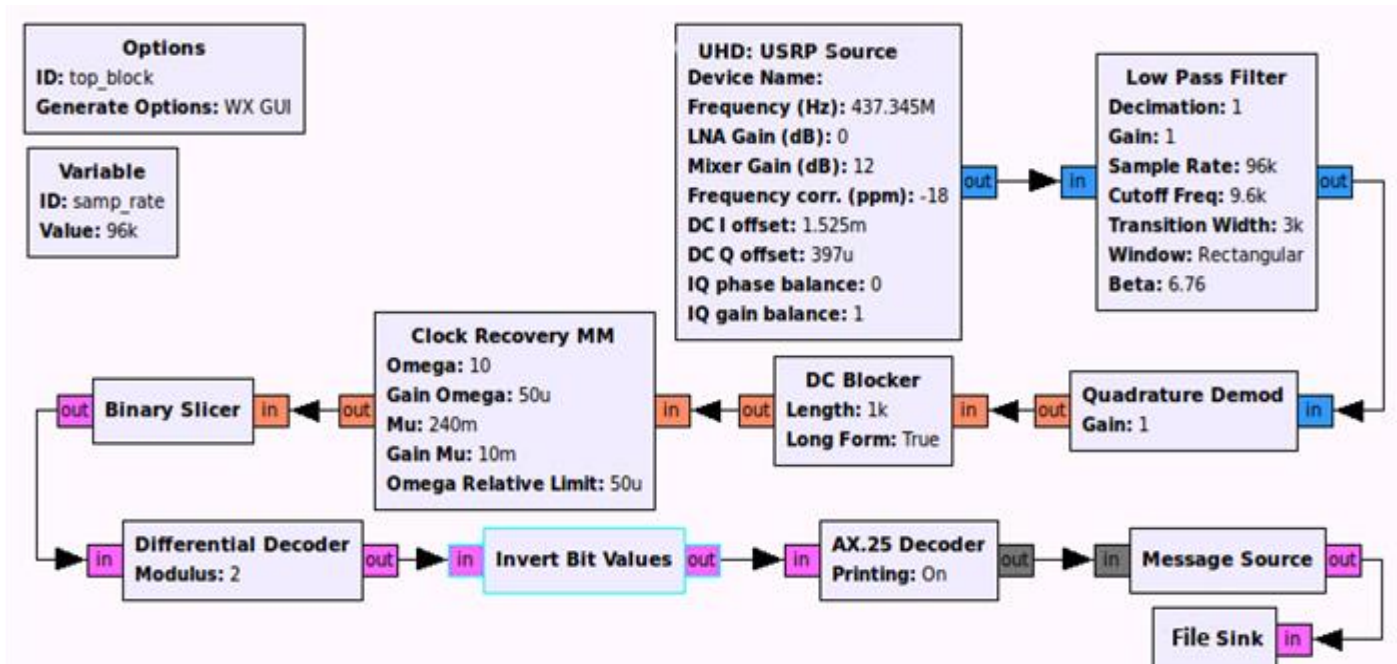


Fig. 6. Implementarea blocului de decodare creat în recepția informație telemetrică.

IV. CONCLUZII

În lucrarea dată s-a relatat conceptul SDR și capabilitățile instrumentarului GNU Radio. SDR oferă o nouă abordare în proiectarea stațiilor terestre dintr-o rețea satelitară, abordare ce aduce cu sine în primul rând o reducere semnificativă a complexității și costurilor de proiectare și oferă un mediu flexibil, versatil de dezvoltare a arhitecturii radio.

Un alt mare avantaj al arhitecturii Software Defined Radio este posibilitatea continuă de a îmbunătăți implementarea software în etape iterative cu corecția ulterioară a erorilor ce apar în timpul funcționării. Totodată, având parametrii de transmisiune de la alți sateliți, fără schimbarea hardware-ului se poate configura echipamentul la parametrii acestora.

Modulele de procesare create (*AX.25 decoder*, *demodulator*) în mediul de dezvoltare GNU Radio, pentru SDR, oferă o serie de posibilități dat fiind faptul că blocurile de decapsulare și decodare bazate pe protocolul AX.25 sunt adaptive pentru comunicare cu alți microsateți ce folosesc acest protocol de comunicare. Totodată modulele date și-ar găsi implementarea într-o rețea de stații terestre pentru comunicare mai eficientă cu microsateți.

Ca direcție de dezvoltare se propune și crearea altor blocuri de procesare pentru mediul GNU Radio, ca de exemplu, blocul de corecție Doppler, care ar avea o dependență directă de datele orbitei satelitului sau alte blocuri de decodare bazate pe diferite protocoale.

Astfel SDR și GNU Radio în special creează o serie de posibilități care depășesc cu mult limitele arhitecturii radio tradiționale

BIBLIOGRAFIE

- [1] Ryan L. Buffington, Craig J. Kief, R. Scott Erwin, Joshua F. Androlewicz and James Lyke, “GENSO, SPA, SDR, and GNU Radio: The Pathway Ahead for Space Dial Tone” American Institute of Aeronautics and Astronautics.
- [2] Matthias Föhnle ” Software-Defined Radio with GNU Radio and USRP/2 Hardware Frontend: Setup and FM/GSM Applications” Hochschule Ulm University of Applied Sciences Institute of Communication Technology Ulm, Germany
- [3] Jan-Hielke Le Roux, “Development of a Satellite Network Simulator Tool and Simulation of AX.25, FX.25 and Hybrid Protocol for Nano-Satellite Communications” Stellenbosch University, October, 2014.
- [4] Chiu-Teng Tsai, Huang-Hua Wen, Yung-Fu Tsai, and Jyh-Chig Juang “Design and Implementation of Microsatellite Telemetry, Tracking and Command (TT&C) Subsystem” Department of Electrical Engineering, National Cheng Kung University No.1, University Road, Taiwan, Journal of Aeronautics, Astronautics and Aviation, Series A, Vol.44, No.1 pp.039, 2012.
- [5] Javier Aguera Reneses, “Development of a Mobile Satellite Ground Communications Station” Universidad Pontificia Comillas, MADRID, 2014.
- [6] Florian George, Stephane Billeter ”AX.25 Telemetry and Telecommand Transfer Frames Format”, Swiss Space Center EPFL, Lausanne, Switzerland, november, 2013.
- [7] Benoit Cosandier, Florian George, Ted Choueiri ”SwissCube Flight Software Architecture” HE-Arc Le Locle, Switzerland, 2007.
- [8] Simona Halunga, Octavian Fratu, Alexandru Vulpe, Răzvan Crăciunescu, Gabriel Toma, Marius Enache ”Echipament radio de emisie recepție scalabil pentru rețele instrumentale de senzori fără fir” Consorțiul SaRaT-IWSN, 2012