



Interfață pentru un comutator rotativ

*Catedra: Calculatoare și Tehnologia
Informației*

Student: Borșa Iulian-Laurențiu
Grupa: 30238
Îndrumător de proiect: Cristi Mocan
Semestrul II, 2019



1. Rezumat	3
2. Introducere	4
2.1. Descrierea problemei de rezolvat	4
2.2. Motivarea proiectului	5
2.3. Organizarea proiectului	5
3. Fundamentare teoretică	6
3.1. Plăci de dezvoltare	6
3.1.1. Digilent Nexys 4	6
3.1.2. Arduino Mega 2560	7
3.2. Microprocesoare și microcontrolere	8
3.2.1. Microprocesor	8
3.2.2. Microcontroler	8
3.3. Tehnologii utilizate	8
3.4. Module și metode	9
3.4.1. Pmod ENC	9
3.4.2. Modul FM	9
4. Proiectare și implementare	10
4.1. Arhitectura sistemului	10
4.2. Algoritmi implementați	11
4.3. Manual de utilizare	11
4.3.1. Conectare plăci	12
4.3.2. Schimbare frecvențe	12
4.3.3. Memorare frecvențe	12
5. Rezultate experimentale	13
5.1. Instrumente de proiectare	13
5.1.1. Limbaj	13
5.1.2. Mediu software	13
5.2. Raport de implementare	13
5.3. Interpretare rezultate	14
6. Concluzii	15
Bibliografie	16
Anexe	16



1. Rezumat

În viața de zi cu zi, utilizăm zilnic comutatoarele rotative atunci când, de exemplu, vrem să ajustăm volumul unor boxe, să reglăm temperatura unei stații de lipit cu cositor sau să schimbăm frecvența unui radio. Proiectul cu "Interfață pentru un comutator rotativ" reprezintă proiectarea unui sistem în care utilizatorul acționează asupra unui buton rotativ. Pentru a se înțelege mai ușor modul în care funcționează un comutator rotativ și rezultatele primite de utilizator să fie cât mai naturale, am asociat proiectul cu un radio. Astfel, am folosit o placă de dezvoltare cu FPGA, o placă de dezvoltare cu microcontroler și câteva module, iar limbajele de programare pe care le-am utilizat au fost C și limbajul de descriere hardware VHDL. Utilizatorul proiectului poate să asculte în timp real radio pe frecvențe cuprinse între 87.5 - 113.0 MHz FM. Frecvența se schimbă acționând comutatorul rotativ, în ambele direcții (trigonometric și în sensul acelor de ceasornic). Sistemul poate memora 16 frecvențe radio găsite de utilizator. Utilizatorul poate să salveze un canal printr-o simplă apăsare a comutatorului.



2. Introducere

2.1. Descrierea problemei de rezolvat

Proiectul "Interfață pentru un comutator rotativ" este o reprezentare a funcționării butoanelor rotative în dispozitivele electrice și electronice. Aceste comutatoare își găsesc utilizarea în diverse domenii, de exemplu, le întâlnim la obiecte pe care le folosim destul de des, precum: boxe, plite electrice, instrumente muzicale, radiouri, stații ș.a.m.d.

În inginerie și nu numai, un comutator este o componentă electrică care are rolul de a permite trecerea curentului electric printr-un circuit. Spre deosebire de comutatoarele normale, care pot doar să închidă și să deschidă un circuit electric, comutatoarele rotative (Rotary Encoder sau Shaft Encoder) sunt dispozitive electro-mecanice care convertesc o poziție a butonului într-un semnal de ieșire analogic sau digital.

Ideea de bază a proiectului constă în realizarea unui sistem care poate fi asociat cu un radio. Mai exact, sistemul are în componență un comutator rotativ. Rotind acest comutator, utilizatorul va putea să schimbe frecvența radio într-un interval standard (87.5 - 113.0 MHz FM), frecvență care se transmite de pe o placă de dezvoltare cu FPGA, la o placă de dezvoltare cu microcontroler Arduino Mega 2560. Mai departe, acest microcontroler va acționa asupra unui modul FM și îi va transmite frecvența. Modulul FM va fi conectat la un sistem audio prin care se vor difuza semnalele radio recepționate.

Sistemul va avea două moduri de funcționare, care constau în căutarea frecvențelor radio și memorarea acestora într-o memorie cu 16 adrese. Pentru a căuta frecvențe radio, utilizatorul va acționa asupra comutatorului rotativ în sensul acelor de ceasornic pentru a crește frecvența și în sens trigonometric pentru a descrește frecvența. De asemenea, utilizatorul poate alege să își salveze frecvențele preferate într-o memorie de 16 adrese, la orice adresă dorește, printr-o simplă apăsare de buton pentru o secundă.

Frecvența la care se ajunge rotind comutatorul va fi afișată pe afișorul cu 7 segmente de pe placă, aceasta fiind cuprinsă în intervalul 87.5 - 113.0 Mhz FM. De la frecvența 113.0 Mhz FM se va face trecerea la 87.5 Mhz FM atunci când se acționează comutatorul rotativ în sensul acelor de ceasornic, iar de la frecvența 87.5 Mhz FM se va face trecerea la 113.0 Mhz FM atunci când se acționează comutatorul rotativ în sens trigonometric. Comutatorul rotativ de pe modul are două funcții. Utilitatea principală a comutatorului este de a schimba starea în funcție de modul în care este rotit, dar în acest proiect poate fi folosit și prin apăsare, pentru a memora. Pentru o funcționare cât mai precisă a comutatorului, 0.1 Mhz FM se vor aduna/scădea la frecvența curentă doar după o rotație completă a comutatorului, adică după traversarea tuturor celor patru stări posibile în care se poate afla codificatorul.

În Figura 2 este reprezentată conectarea componentelor sistemului, iar componentele sistemului sunt:

- placă de dezvoltare cu FPGA Nexys4
- placă de dezvoltare cu microcontroler Arduino Mega 2560
- comutator rotativ de pe Digilent Pmod ENC
- modul FM RRD-102
- breadboard, fire și boxe

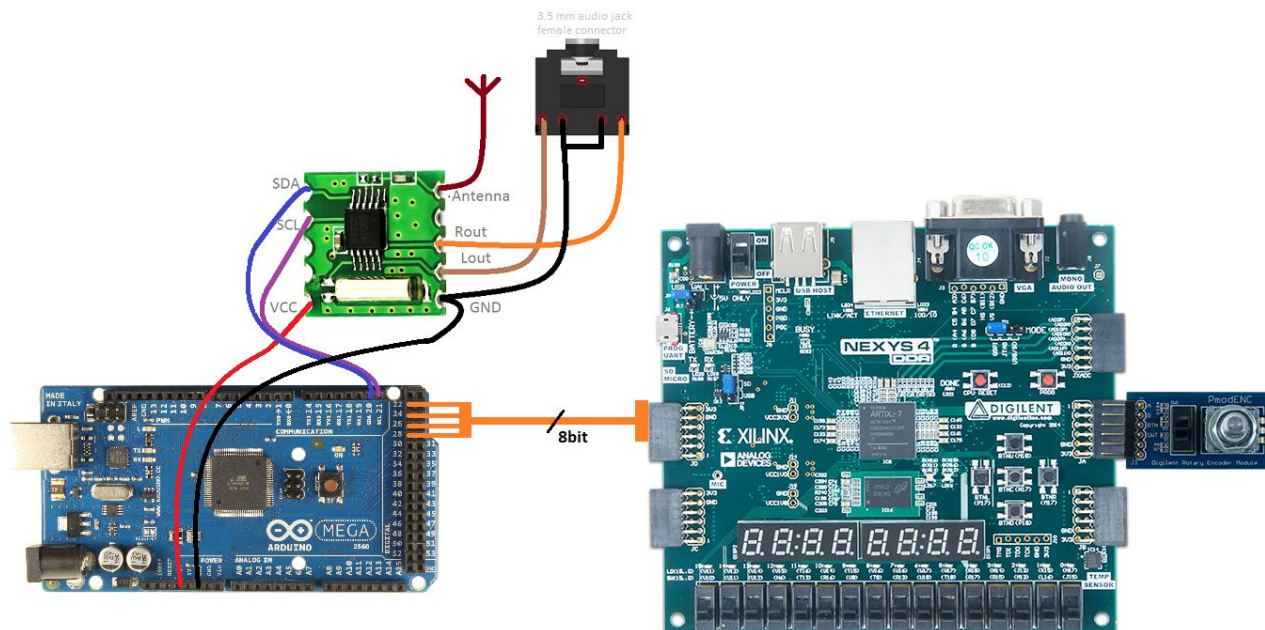


Figura 2.1 - Componentele sistemului

2.2. Motivarea proiectului

Principala motivație a proiectului o constituie proiectarea unui sistem care să reprezinte o interfață cu un comutator rotativ, dar care să aibă o utilitate în viața de zi cu zi. Astfel, am ales să proiectez un radio folosind două plăci de dezvoltare, două module și conectori. În așa fel, o motivație secundară este aceea de a utiliza în același ansamblu componente care se programează în limbaje de descriere sau programare diferite. În acest proiect sunt îmbinate implementările software și hardware. Utilizatorul acționează direct pe sistemul hardware, care preia informația (datele de intrare) pe care o prelucrează ulterior într-un sistem software, urmând ca apoi să trimită rezultatele tot la sistemul hardware.

2.3. Organizarea proiectului

În secțiunea “Fundamentare teoretică” sunt descrise fundamentele teoretice legate de proiect, cum sunt modulele, metodele și tehnologiile utilizate. Toate componentele sunt documentate separat, la fel și principiile de funcționare a modulelor.

Secțiunea “Proiectare și implementare” prezintă implementarea sistemului cu explicarea comportamentului fiecărui modul VHDL, cu interconectarea componentelor și parcursul semnalelor și al datelor de la intrare până la ieșire. Vor fi explicate diagrame și scheme bloc.

În secțiunea “Rezultate experimentale” sunt analizate rapoartele de implementare și se interpretează rezultatele obținute după încărcarea codului pe plăci.

Secțiunea “Concluzii” prezintă sumarul proiectului, contribuțiile sistemului propus și posibilități de dezvoltare ulterioare. În final sunt listate referințele bibliografice și anexele care conțin codul sursă.



3. Fundamentare teoretică

3.1. Plăci de dezvoltare

3.1.1. Digilent Nexys 4

Placa Nexys 4 este o platformă de dezvoltare bazată pe FPGA (Field Programmable Gate Array) Artix-7™ de la Xilinx. Nexys 4 este compatibil cu noua suită de design Vivado® Design Suite de la Xilinx, precum și cu instrumentele ISE, care includ ChipScope și EDK. Xilinx oferă pachete gratuite “Webpack” ale acestor seturi de instrumente, astfel încât proiectele să poată fi implementate fără costuri suplimentare. Placa conține un singur oscilator cu frecvența de ceas de 100MHz conectat la pinul E3. Se pot folosi circuite adiționale pentru a diviza frecvența astfel încât să fie potrivită pentru toate componentele.

Placa Nexys 4 include un FTDI FT2232HQ USB-UART bridge (atașat la pinul J6) care permite programatorului să folosească aplicațiile dezvoltate în mediul software Xilinx să comunice direct cu placa prin standardul Windows COM Port.

Nexys4 dispune de 16 diode emițătoare de lumină (LED) și de două afișoare cu 7 segmente pentru 4 cifre. Display-ul de 7 segmente (de asemenea denumit, afișaj LED cu 7 segmente) este o modalitate de a reprezenta numere în sisteme electronice. Acesta este compus din 7 segmente care pot fi activate sau dezactivate individual. Fiecare segment are forma unei linii scurte. Se poate compara cu scrierea de numere cu ajutorul bețelor de chibrit. Fiindcă o cifră este formată din 7 LED-uri și pe Digilent Nexys 4 avem 8 cifre, în mod normal am avea nevoie de 40 de biți pentru a controla starea fiecărui LED (anod + catod). Totuși este folosită o schema de multiplexare prin care un anod este comun tuturor celor 7 LED-uri dintr-o cifră, iar catodii LED-urilor de pe aceeași poziție sunt partajați între toate cele 8 cifre. Folosind o schemă de multiplexare, doar o cifră poate fi aprinsă la un moment dat (dacă se dă drumul la mai mult de un anod aceeași cifră va fi afișată pe toate pozițiile, deoarece catodii sunt partajați). Astfel, dacă dorim să afișăm mai multe cifre diferite între ele, nu putem face acest lucru decât afișând câte una la singură la fiecare moment de timp. Din fericire, lucrând cu un impuls de ceas intern de frecvență cu mult mai mare decât capacitatea de percepție a ochiului uman, ne permitem să facem un astfel de afișaj rând-pe-rând, urmând ca ochii noștri să perceapă un display continuu al celor 8 cifre.

Porturile Pmod sunt aranjate în câte două rânduri de 6 pini, dintre care 4 sunt de date și doi sunt pentru 3.3V VCC (pinii 6 și 12) și GND (pinii 5 și 11), ca în Figura 3.1.1.c.

Ultima componentă folosită de pe placă este butonul CPU Reset, care are rolul de a reseta programul încărcat pe placă. Butonul roșu CPU Reset generează o ieșire high când nu este acționat și o ieșire de low când este apăsat.

Anterior au fost menționate toate intrările și ieșirile pe care le posedă Nexys 4 și care sunt folosite în proiect. Pe lângă acestea, plăcile Nexys mai au 16 comutatoare, 5 butoane, slot pentru memorie SD, LED-uri multicolor, porturi VGA, câțiva senzori integrați pentru accelerometru sau senzor de temperatură. În plus, mai poate fi folosită și ieșirea audio - stereo sau microfonul.

În cele ce urmează sunt prezentate schemele celor patru componente utilizate de pe placa de dezvoltare Nexys 4.

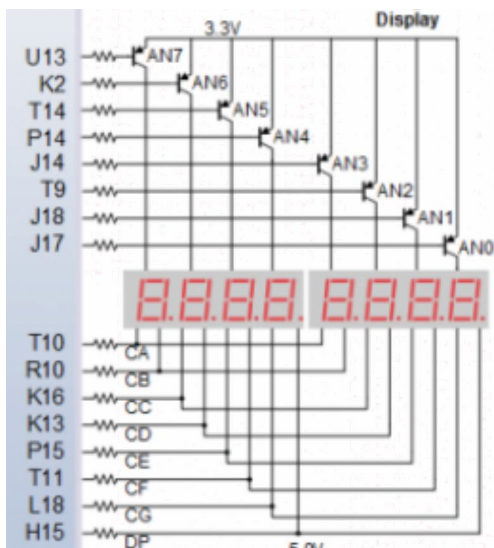


Figura 3.1.1.a - Schema afișorului

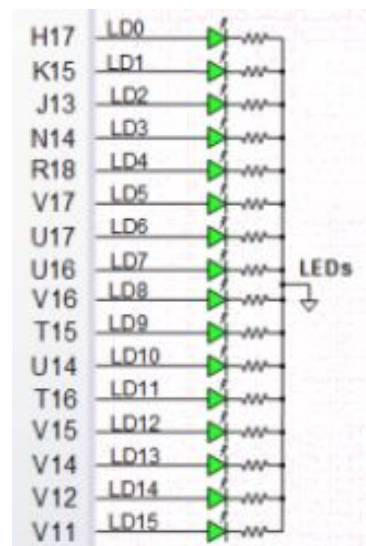


Figura 3.1.1.b - Schema diodelor



Figura 3.1.1.c - Schema Pmod



Figura 3.1.1.d - Schema CPU Reset

3.1.2. Arduino Mega 2560

Arduino Mega 2560 este o placă de dezvoltare cu microcontroller bazată pe ATmega2560. Aceasta are 54 de pini digitali de intrare/ieșire, dintre care 14 pot fi folosiți ca ieșiri PWM, 16 intrări analog, 4 porturi seriale UART, un oscilator de quartz de 16 MHz, o conexiune USB, un jack pentru alimentare direct de la priză și un buton de reset. Aceasta conține orice este nevoie pentru a suporta microcontrollerul. În figura de mai jos este prezentată placa cu I/O.

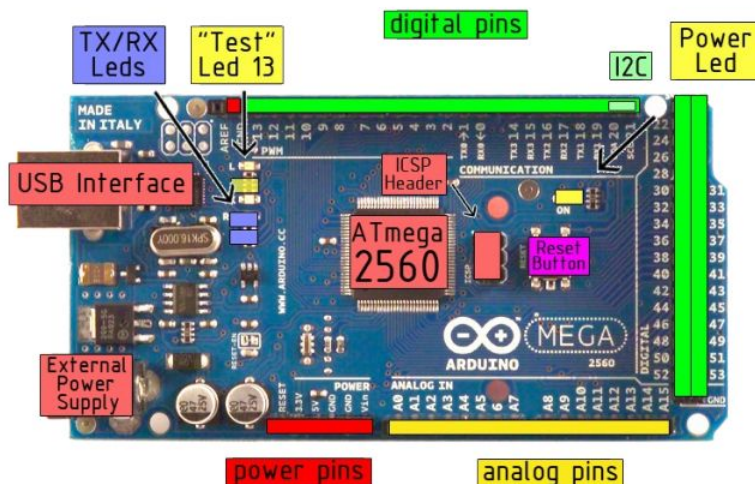


Figura 3.1.2 - Harta pinilor Arduino Mega 2560



3.2. Microprocesoare și microcontrolere

Microcontrolerul diferă de un microprocesor în multe feluri. În primul rând și cel mai important este funcționalitatea sa. Pentru a fi folosit, unui microprocesor trebuie să i se atașeze alte componente ca memorie și componente pentru primirea și trimiterea de date.

3.2.1. Microprocesor

Microprocesorul, uneori numit și procesor, este unitatea centrală de prelucrare a informației (CPU) a unui calculator sau sistem structurat funcțional, care coordonează sistemul și care, fizic, se prezintă sub forma unui cip electronic. UCP conține o structură de procesare (numită „procesor”) alcătuită din:

- Unitate de control, care decodifică instrucțiunile binare ale unui program, generează semnalele de control pentru celelalte unități funcționale ale procesorului, generează semnalele de control pentru componentele externe ale procesorului: memorie externă, sistem de intrare ieșire
- Unitate aritmetică și logică: unitate de prelucrare a datelor, funcția specifică executată fiind stabilită în fiecare moment prin semnalele de control primite de la UC
- Registre de uz general folosite ca memorie locală de mare viteză, magistrale interne pentru transferul datelor, adreselor și instrucțiunilor

3.2.2. Microcontroler

Microcontrolerul este o structură electronică destinată controlului unui proces sau, mai general, este un microcircuit care încorporează o unitate centrală (CPU) și o memorie împreună cu resurse care-i permit interacțiunea cu mediul exterior.

3.3. Tehnologii utilizate

VHDL este utilizat în mod frecvent pentru simularea design-ului electronic și sinteza acestora. Sinteza este procesul în care sursa VHDL este compilată și mapată în tehnologii de genul FPGA (Field Programmable Gate Array).

FPGA este un circuit integrat digital configurabil. Funcțiile dispozitivelor pot fi configurate, de către utilizator, în orice moment după fabricare. FPGA este un tip de circuit logic programabil. Configurarea FPGA se face, în general, cu ajutorul unui limbaj de descriere hardware HDL (Verilog sau VHDL). FPGA-urile sunt alcătuite din blocuri logice configurabile (programabile) legate între ele de o serie de conexiuni configurabile la rândul lor. Mai jos este prezentată schema unui bloc logic de FPGA.

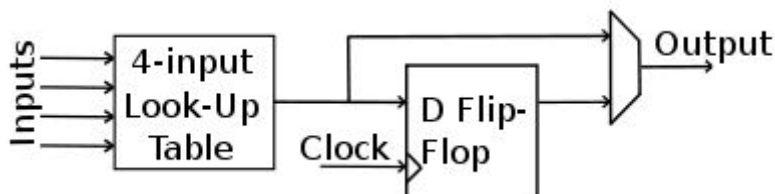


Figura 3.3.1 - Schema unui bloc logic la FPGA



3.4. Module și metode

3.4.1. Pmod ENC

Modulul Pmod ENC utilizează un comutator rotativ Rotary Encoder ca o cale pentru utilizatori pentru a comuta rapid între opțiuni multiple, precum alegerea luminozității unui ecran, setarea volumului unui sistem audio sau pentru a alege viteza unui motor. Acest modul dispune, pe lângă comutatorul rotativ, care poate să fie apăsăat și să funcționeze ca un buton, de un comutator cu două stări.



Figura 3.4.1.a - Modul Pmod ENC

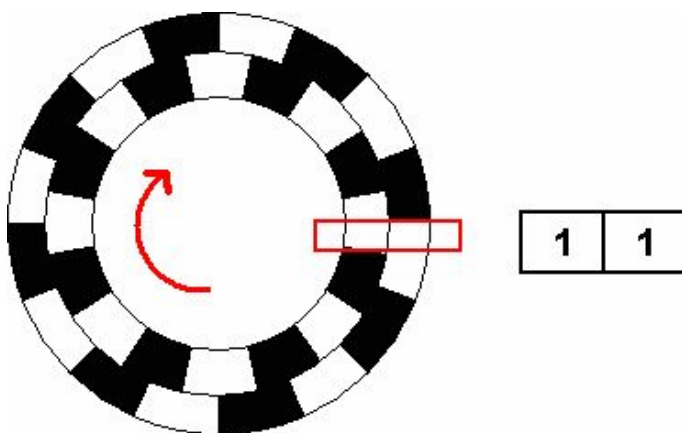


Figura 3.4.1.b - Codificare pentru Pmod ENC

3.4.2. Modul FM

Modul pentru Radio FM cu circuit integrat RDA5807M care poate să fie controlat de un microcontroller sau plăcuțe de dezvoltare Arduino. Acest modul este potrivit pentru a fi integrat în proiecte care constau în crearea unui radio. De exemplu, cu acest modul se poate construi propriul sistem audio, iar modulul Radio face parte din el. În figurile de mai jos este prezentat Modulul FM și schema pinilor pentru conectarea acestuia la placa de dezvoltare Arduino Mega 2560 și la sistemul audio.



Figura 3.4.2.a - Modul FM



Figura 3.4.2.b - Schema pinilor pentru Modul FM



4. Proiectare și implementare

4.1. Arhitectura sistemului

În figura 4.1 este prezentată schema bloc a arhitecturii generale a sistemului implementat. Porturile de I/O pentru Nexys 4 sunt următoarele:

- Reset - in, direct de la butonul de reset de la placă
- JB - in, de la un port Pmod, direct de la Pmod ENC
- Anod - out, direct la anodul afișorului cu 7 segmente de la placă
- Segment - out, direct la catodul afișorului cu 7 segmente de la placă
- JA - out, la un port Pmod, conectat la Arduino Mega 2560
- Led - out, direct la diodele luminiscente de la placă

Semnalele de comandă pentru porturile de I/O au exact același nume cu porturile prezentate mai sus.

Toate aceste porturi fac posibilă interfața cu utilizatorul.

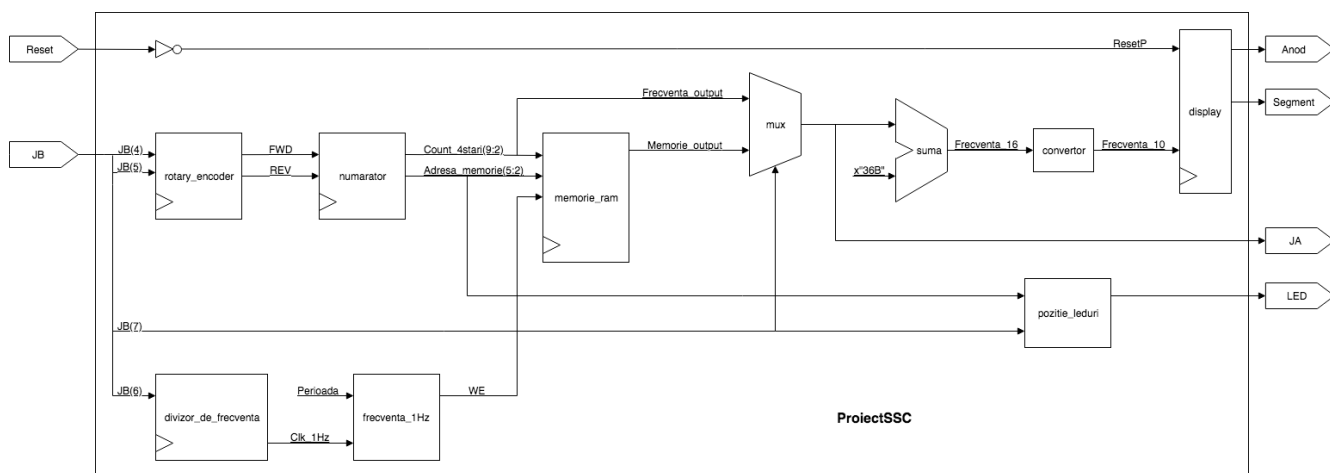


Figura 4.1 - Schema bloc a arhitecturii sistemului

Arhitectura principală este compusă din cinci componente, dintre care:

- componenta responsabilă de afișarea frecvenței radio pe afișorul cu 7 segmente
- componenta care transformă un număr din baza 16 în baza 10
- componenta care interpretează semnalele primite de la comutatorul rotativ
- divizorul de frecvență
- memoria RAM



4.2. Algoritmi implementați

Algoritmul pe care se bazează acest proiect este cel din componenta care prelucrează semnalele de la comutatorul rotativ. În principiu, pentru a crește sau descrește frecvența dispozitivului radio cu 0.1 MHz FM, comutatorul va trece prin patru stări. Pentru a face mișcarea mai precisă și mai apropiată de funcționarea reală a unui comutator rotativ, am hotărât să fac aceste operații după patru stări consecutive. Mai exact, dacă starea inițială este x"00", pentru a face o creștere sau descreștere, comutatorul va trece prin x"01", x"11" și x"10" (depinde de sens).

4.3. Manual de utilizare

În Figura 4.3 este prezentată interfața pentru comutatorul rotativ. Figura conține doar placa de dezvoltare Nexys4 și modulul Pmod ENC deoarece acestea sunt singurele componente de pe care se poate controla proiectul.

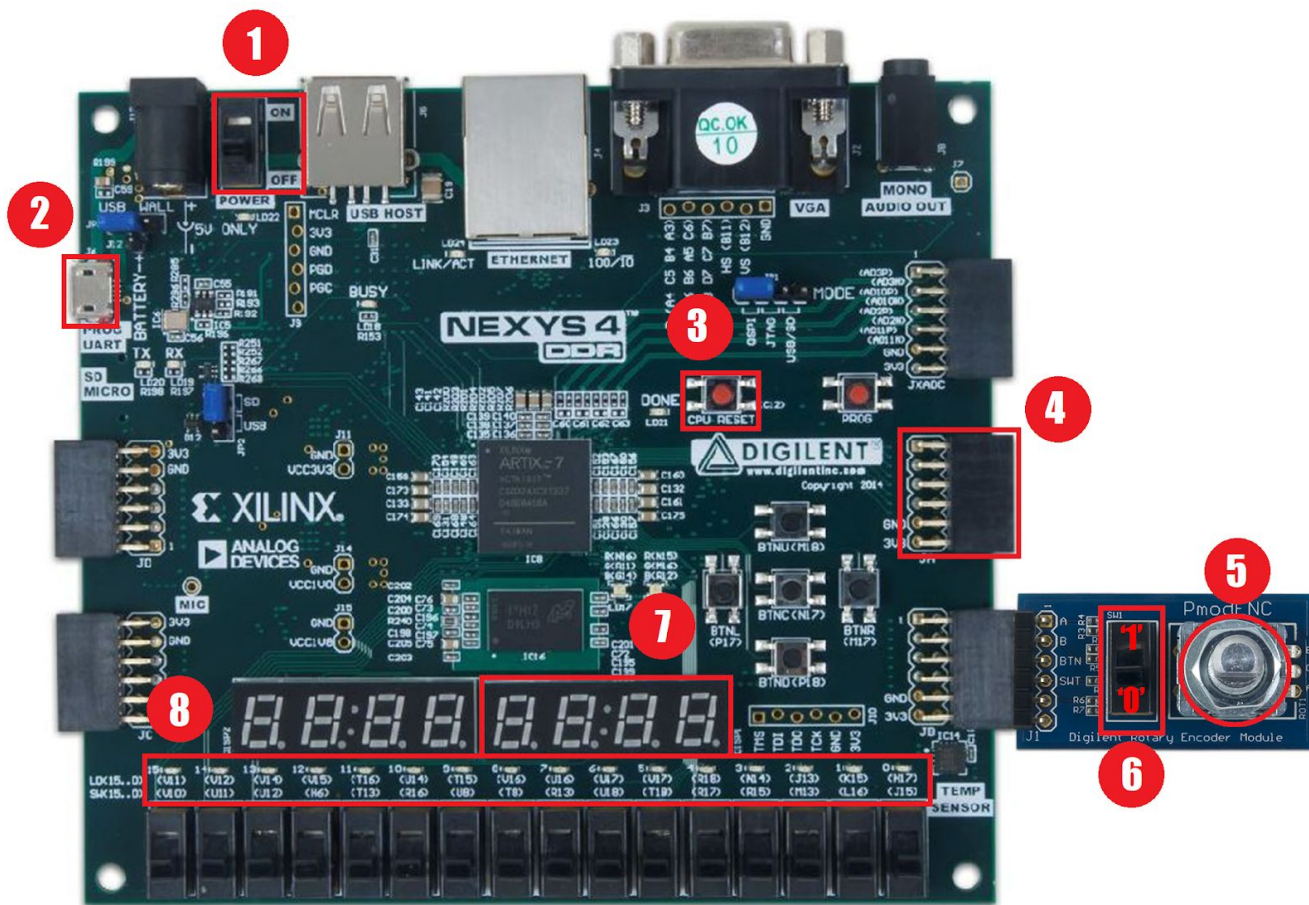


Figura 4.3 - Ghid de utilizare



Tabel 4.3 - Descrierea componentelor

Număr	Denumire
1	Comutator pornire/oprire placă
2	Port UART USB
3	Buton de reset
4	Conector Pmod către Arduino
5	Comutator rotativ
6	Switch pentru mod
7	Afișor cu 7 segmente
8	Leduri

În continuare este prezentat modul de funcționare al proiectului:

4.3.1 Conectare plăci

- Se conectează un cablu cu mufă USB-Micro-B la placa Nexys4 în intrarea (2)
- Se alimentează placa Arduino cu un cablu cu mufă USB-B sau DC la 12V
- Pentru pornire, se trece comutatorul (1) în poziția ON, iar pentru oprire în poziția OFF
 - Afișorul (7) va afișa rezultatul 0 | 8 | 7 | 5, care reprezintă frecvența de start
 - Dacă comutatorul (6) este în poziția '0', sistemul se află în modul de căutare a frecvențelor, ledurile (8) sunt stinse, iar prin rotirea comutatorului (5) se schimbă frecvența radio trimisă prin portul (4) la Arduino
 - Dacă comutatorul (6) este în poziția '1', sistemul se află în modul de memorare a frecvențelor, câte unul dintre ledurile (8) este aprins, iar prin rotirea comutatorului (5) se schimbă adresa de memorie în care se stochează frecvența radio trimisă prin portul (4) la Arduino

4.3.2 Schimbare frecvențe

- Se trece comutatorul (6) în poziția '0'
- Se rotește comutatorul (5) în sensul acelor de ceasornic pentru creșterea frecvenței și în sens trigonometric pentru descreșterea frecvenței
 - Afișorul (7) va afișa frecvența la care s-a ajuns
 - Ledurile (8) sunt stinse
 - Prin apăsarea timp de o secundă a comutatorului (5) se reține frecvența în memoria RAM



4.3.3 Memorare frecvențe

- Se trece comutatorul (6) în poziția '1'
- Se rotește comutatorul (5) în sensul acelor de ceasornic pentru creșterea adresei de memorare și în sens trigonometric pentru descreșterea adresei de memorare
- Se apasă timp de o secundă comutatorul (5) pentru a reține frecvența în memoria RAM la poziția indicată pe ledul (8)
 - Afișorul (7) va afișa inițial rezultatul `| 0 | 8 | 7 | 5 |`, care reprezintă valoarea standard înregistrată în memorie de către programator, dacă nu s-a memorat o altă valoare pe poziția respectivă
 - Unul dintre ledurile (8) este aprins și indică adresa memoriei unde se va face următoarea stocare de frecvență

5. Rezultate experimentale

5.1. Instrumente de proiectare

5.1.1 Limbaj

Pentru elaborarea proiectului, am folosit două limbaje diferite. Pentru programarea plăcii de dezvoltare Nexys4 am folosit limbajul de descriere hardware VHDL, iar pentru programarea plăcii cu microcontroler Arduino Mega 2560 am folosit extensia limbajul de programare C/C++, Embedded C.

5.1.2 Mediu software

Pentru a putea programa ambele plăci, am folosit două medii software diferite. Pentru VHDL am folosit Xilinx Vivado Design Suite 2018.3, iar pentru Embedded C am folosit Arduino IDE. Sistemul de operare de pe care am programat este Windows 10 Pro.

5.2. Raport de implementare

Mediul de proiectare Xilinx Vivado Design Suite generează statistici pentru proiecte. Mai jos sunt prezentate câteva rapoarte generate de mediul software. În Figura 5.2 este reprezentat un grafic cu procentajul de LUT (LookUp Table), Bistabile (FF) și I/O folosite pentru proiect. Un LUT, care reprezintă LookUp Table, în termeni generali este în esență un tabel care determină ce este ieșirea pentru o anumită intrare de date. În contextul logicii combinaționale, este tabelul adevărului. Acest tabel de adevăr definește modul în care se comportă logica combinațională. Tabelul 5.2 afișează numărul resurselor utilizate, prezentate în Figura 5.2 ca procentaje.



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

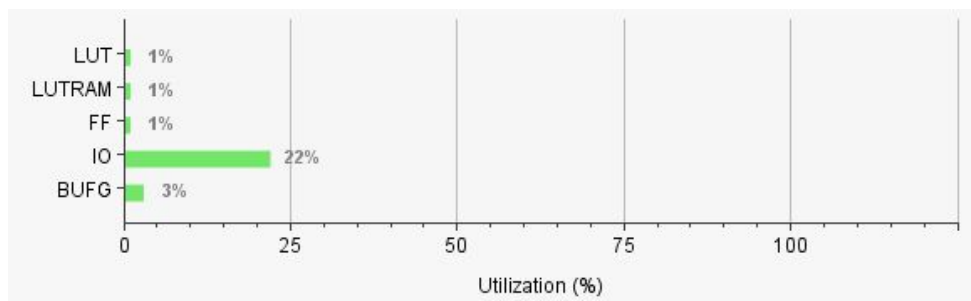


Figura 5.2 - Grafic cu procentajul resurselor utilizate

Tabel 5.2 - Tabel cu numărul resurselor utilizate

Resource	Utilization	Available	Utilization %
LUT	712	63400	1.12
LUTRAM	8	19000	0.04
FF	76	126800	0.06
IO	46	210	21.90
BUFG	1	32	3.13

5.3. Interpretare rezultate

În continuare sunt prezentate două imagini cu sistemul funcționând în ambele moduri:

În Figura 5.3.a este prezentată o imagine din modul de memorare, înainte de înregistrarea unei frecvențe noi. S-a rotit de comutator până la poziția în care ledul aprins reprezintă adresa a 11-a de memorare.

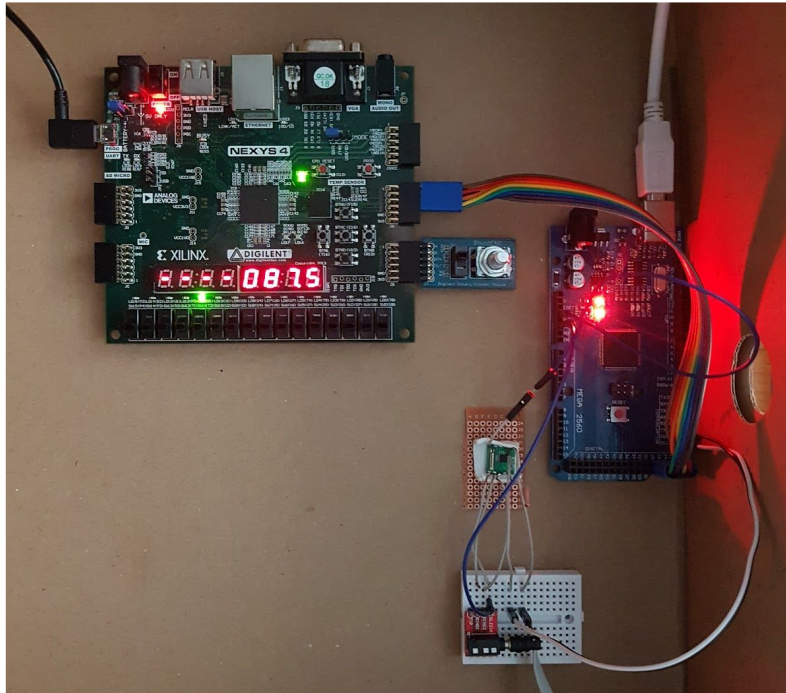


Figura 5.3.a - Mod memorare

În Figura 5.3.b este prezentată o imagine din modul de căutare. S-a rotit de comutator până la frecvența 89.8fm.

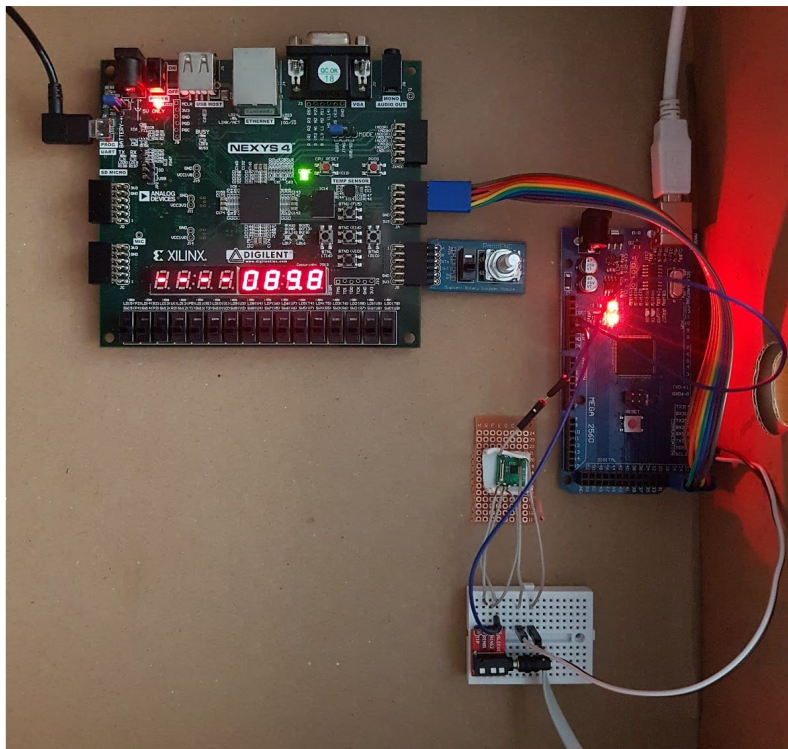


Figura 5.3.b - Mod căutare



6. Concluzii

Problema rezolvată de proiect constă în utilizarea unui comutator rotativ folosind o placă de dezvoltare cu FPGA Nexys4 și un modul Pmod ENC. Obiectivul principal al proiectului a fost acela de a prelua date de la comutator în timpul rotirii acestuia. Pentru a face proiectul mai interactiv, un al doilea obiectiv a fost legarea împreună, în același sistem, a unei plăci de dezvoltare cu FPGA și a unei plăci de dezvoltare cu microcontroler Arduino.

Contribuția originală la acest proiect este punerea în comun, într-un singur sistem, a două plăci de dezvoltare care necesită limbaje de programare/descriere hardware diferite, cu câte un modul pentru fiecare placă. În altă ordine de idei, în afară de implementarea modului pentru comutatorul rotativ, tot comportamentul proiectului are o contribuție originală, deoarece am extins cerința inițială.

Avantajele proiectului:

- viteză mare de procesare
- ușor de realizat
- cod puțin, simplu și ușor de înțeles

Dezavantajele proiectului:

- preț mare al componentelor
- nu toate componentele sunt realizate optim

O aplicație a proiectului de bază "Interfață pentru un comutator rotativ" e reprezentată de produsul final, mai exact, de acest radio. Alte sugestii ar putea fi: comutator rotativ pentru electrocasnice, comutator rotativ pentru diverse acțiuni la jocuri, volum pentru boxe.

Ca și dezvoltări ulterioare, proiectul cu radio ar putea avea o opțiune pentru setarea volumului, afișarea frecvenței, a volumului și a locației de memorie pe un display LCD. De asemenea, alte dezvoltări ulterioare ar putea fi transformarea componentelor realizate comportamental, în componente descrise prin flux de date sau structural. Astfel se optimizează sistemul.

Bibliografie

- [1] Nexys4™ FPGA Board Reference Manual (Nexys4 rev. B; Revised September 6, 2013). https://www.xilinx.com/support/documentation/university/Vivado-Teaching/HDL-Design/2013x/Nexys4/Supporting%20Material/Nexys4_RM_VB1_Final_3.pdf
- [2] "Rotary encoder", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Rotary_encoder
- [3] Arduino Mega 2560, <http://roboromania.ro/datasheet/Arduino-Mega-2560-roboromania.pdf>
- [4] M. Hertel, "mathertel/Radio", GitHub, 2019. [Online]. Available: <https://github.com/mathertel/Radio/blob/master/src/RDA5807M.h>
- [5] "VHDL rotary encoder interface", All About Circuits, 2019. [Online]. Available: <https://forum.allaboutcircuits.com/threads/vhdl-rotary-encoder-interface.23344/>
- [6] A. Aqeel, "Introduction to Arduino Mega 2560 - The Engineering Projects", The Engineering Projects, 2019. [Online]. Available: <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-mega-2560.html>



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

[7] I. Dogaru and R. Dogaru, "Curs Sisteme Reconfigurabile de Calcul", Univ. Politehnica din Bucuresti, 2016. [Online]. Available:

http://atm.neuro.pub.ro/radu_d/html/09_10/src2009/1/draft_curs5.pdf

[4] "FPGA", Ro.wikipedia.org. [Online]. Available: <https://ro.wikipedia.org/wiki/FPGA>

[5] M. Romanca, "Microprocesoare și microcontrolere", Universitatea Transilvania din Brașov, 2015. [Online]. Available:

<http://vega.unitbv.ro/~romanca/Carte-MpMc%202015/Microprocesoare%20si%20microcontrolere-978-606-19-0683-3.pdf>



Anexe