



**UNIVERSITATEA
TEHNICĂ
DIN CLUJ-NAPOCA**

**PROIECT STRUCTURA SISTEMELOR
DE CALCUL**

Toader Eric-Stefan si Pupeza Bogdan Nicolae

Elaborarea unui proiect folosind limbajul de
proiectare hardware VHDL

2022/2023

Profesori indrumatori: Filip Gherman, Florin Lisman

Cuprins

Rezumat.....	2
Introducere.....	2
Fundamentare teoretica	3
Proiectare si implementare	4
Concluzie	9
Bibliografie.....	9

Rezumat

Proiectul acesta a presupus măsurarea temperaturii cu senzorul de pe placa Nexys 4 DDR și transmiterea ei la un dispozitiv mobil prin intermediul unui modul Bluetooth.

Cerintele proiectului în integritate au fost:

- Proiectare utilizând limbajul VHDL
- Implementare pe placa Digilent Nexys 4 DDR
- Măsurarea temperaturii cu senzorul de temperatură ADT7420
- Utilizarea modului Bluetooth Pmod BT2 pentru transmiterea temperaturii la un dispozitiv mobil
- Aplicație pe dispozitivul mobil pentru preluarea și afișarea temperaturii

Am reușit să transmitem temperatura ambientală citită prin intermediul senzorului de temperatură prezent pe placa Nexys 4 DDR interfatându-l pe acesta cu o componentă proiectată conform documentației Digilent. Această temperatură a fost mai apoi transmisă prin componenta transmitătoare Bluetooth, care efectuează transmisii recurente doar atunci când temperatura se schimbă.

Introducere

Proiectul prezentat în acest raport se concentrează pe utilizarea limbajului VHDL pentru a construi un sistem capabil să citească valorile de temperatură de pe senzorul integrat al plăcii Nexys 4 DDR și să le trimită prin Bluetooth folosind modulul Pmod BT2. Acest sistem poate fi setat să trimită valorile automat sau doar la apăsarea unui buton. Senzorul integrat de temperatură utilizat în acest proiect este un dispozitiv electronic care poate detecta și converti variațiile de temperatură în semnale electrice, care pot fi apoi prelucrate de sistemul proiectat. Acest senzor este o componentă importantă pentru monitorizarea și controlul temperaturii în diferite aplicații.

Modulul Pmod BT2 este un dispozitiv de comunicație wireless care permite transmiterea de date prin intermediul tehnologiei Bluetooth. Acest modul este utilizat pentru a transmite valorile de temperatură citite de senzor către alte dispozitive conectate, astfel permițând monitorizarea și controlul temperaturii de la distanță.

În domeniul tehnologiei IoT, monitorizarea și controlul temperaturii devine din ce în ce mai importantă, deoarece poate fi utilizată în aplicații precum managementul energiei sau în industria alimentară. Problema de rezolvat în acest proiect este construirea unui sistem capabil să citească valorile de temperatură de pe un senzor specific și să le transmită prin Bluetooth. Obiectivele principale ale proiectului sunt: implementarea unui sistem eficient de citire a temperaturii, transmiterea acestor valori prin Bluetooth folosind modulul Pmod BT2 și posibilitatea de a seta modul de transmitere automat sau la apăsarea unui buton.

Această soluție se diferențiază de altele prin posibilitatea de a seta modul de transmitere automat sau la apăsarea unui buton.

Raportul va conține informații despre implementarea sistemului, testarea acestuia și concluziile trase în urma proiectului.

Fundamentare teoretică

UART, sau Universal Asynchronous Receiver Transmitter, este un circuit integrat care permite transmiterea de date între două sisteme prin intermediul unei linii seriale. UART convertește semnalele digitale în semnale analogice pentru a fi transmise prin intermediul unei linii seriale, și invers, convertind semnalele analogice în semnale digitale pentru a fi prelucrate de sistem.

Bluetooth UART (Universal Asynchronous Receiver-Transmitter) este un protocol de comunicație care permite transmiterea de date între două dispozitive Bluetooth prin intermediul unei linii seriale. Acest protocol este similar cu UART standard, cu diferența că datele sunt transmise prin intermediul unei conexiuni wireless Bluetooth în loc de o linie serială fizică.

În cazul Bluetooth UART, transmiterea de date se realizează prin intermediul unui pin RX (receiver) și un pin TX (transmitter) de pe modulul Bluetooth utilizat. Pinul RX este utilizat pentru a primi datele transmise de dispozitivul conectat, iar pinul TX este utilizat pentru a transmite datele către dispozitivul conectat.

Atunci când un dispozitiv transmite date către alt dispozitiv, acestea sunt convertite în semnale digitale și apoi transmise prin intermediul pinului TX. Acestea sunt apoi recepționate de către pinul RX al dispozitivului conectat, unde sunt convertite din nou în semnale digitale pentru a fi prelucrate de sistem.

Senzorul de temperatură utilizat în acest proiect este ADT7420, un senzor de temperatură cu precizie ridicată și un răspuns rapid. Acesta are o plajă de măsurare între -40°C și $+125^{\circ}\text{C}$ și este capabil să furnizeze temperatura cu o precizie de $\pm 0.5^{\circ}\text{C}$ în plajă -40°C to $+85^{\circ}\text{C}$ și cu o precizie de $\pm 1^{\circ}\text{C}$ în plajă -40°C to $+125^{\circ}\text{C}$. Senzorul ADT7420 poate fi citit prin intermediul unei interfețe I2C sau SPI.

Pentru citirea temperaturii din senzorul ADT7420, se va utiliza o interfață I2C sau SPI, aceasta va fi utilizată pentru a comunica cu senzorul și a citi valorile de temperatură. Acest proces va implica transmiterea unor comenzi și citirea unor date de la senzor, care vor fi prelucrate și utilizate pentru a afișa valorile de temperatură.

Proiectare si implementare

Implementarea hardware a acestui proiect include următoarele etape:

1. Conectarea modului Pmod BT2 la placa Nexys 4 DDR: Acest lucru poate fi realizat prin intermediul conectorilor UART sau prin utilizarea unui cablu de legătură.
2. Scrierea codului VHDL pentru a controla senzorul de temperatură și modulul Pmod BT2: Acest cod ar trebui să permită citirea valorilor de temperatură de pe senzor și transmiterea acestor valori prin intermediul modului Pmod BT2.
3. Implementarea codului VHDL pe placa Nexys 4 DDR: Acest lucru poate fi realizat prin intermediul unui programului Vivado.
4. Testarea sistemului: După ce sistemul este implementat, ar trebui să fie testat pentru a se asigura că senzorul de temperatură poate citi valorile corect și că acestea pot fi transmise prin intermediul modului Pmod BT2.
5. Îmbunătățirea sistemului: Dacă sunt descoperite erori sau probleme în timpul testării, ar trebui să fie realizate modificări pentru a îmbunătăți performanța sistemului.
6. Conectarea unui dispozitiv mobil pentru a receptiona valorile de temperatura transmise prin Bluetooth si afisarea lor.

Motivarea solutiei alese:

Există mai multe motive pentru care UART este o alegere potrivită pentru acest proiect:

1. Simplitate: UART este un protocol de comunicare simplu care poate fi implementat cu ușurință pe o varietate de plăci de dezvoltare.
2. Viteză: UART poate furniza o viteză de transmisie mare, ceea ce este important pentru transmiterea valorilor de temperatură în timp real.
3. Fiabilitate: UART este un protocol de comunicare stabil și fiabil, ceea ce înseamnă că este puțin probabil ca datele să se piardă sau să fie corupte în timpul transmiterii.
4. Suport pentru mai multe plăci: UART este suportat de majoritatea plăcilor de dezvoltare, ceea ce înseamnă că poate fi utilizat cu ușurință în combinație cu placa Nexys 4 DDR.
5. Cost redus: UART este un protocol de comunicare ieftin care poate fi implementat cu un cost redus.

Toate acestea fac ca UART să fie o alegere potrivită pentru acest proiect, deoarece necesită un efort minim pentru implementare, oferă viteză și fiabilitate în transmisie, și este suportat de majoritatea plăcilor de dezvoltare.

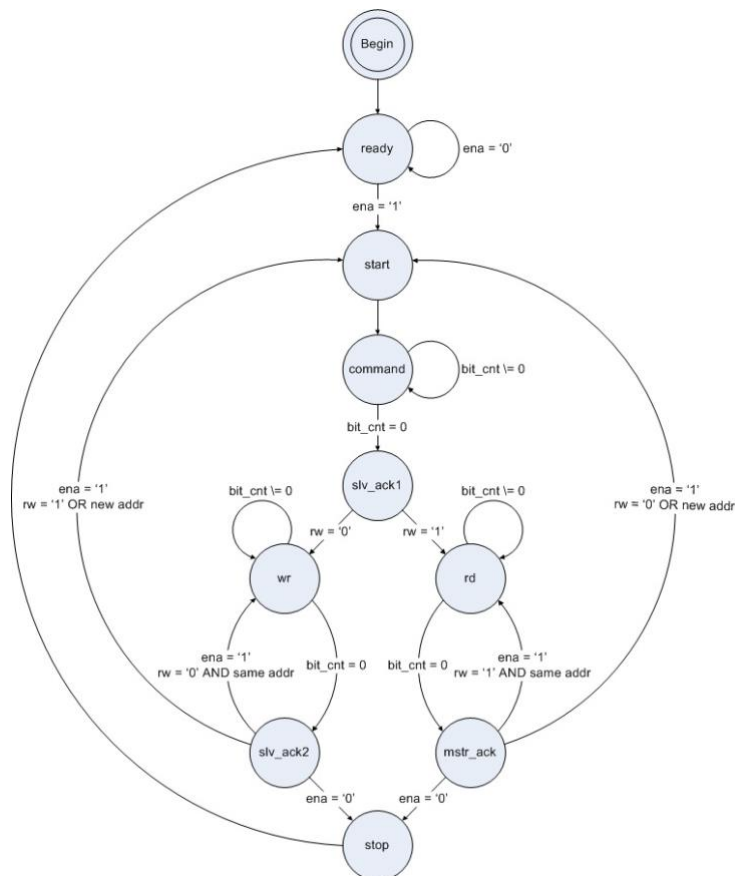
I2C este ales pentru acest proiect deoarece oferă următoarele avantaje:

1. Multi-master: I2C permite mai mult de un dispozitiv să fie master în rețea, ceea ce permite mai multe dispozitive să acceseze un bus comun, ceea ce face proiectul pregatit pentru dezvoltari ulterioare.
2. Consum redus de energie: I2C are un consum redus de energie comparativ cu alte protocoale de comunicare, ceea ce poate fi benefic pentru aplicațiile cu limitări de alimentare.
3. Fiabilitate: I2C este un protocol de comunicare stabil și fiabil, ceea ce înseamnă că este puțin probabil ca datele să se piardă sau să fie corupte în timpul transmiterii.

Implementarea modului I2C:

1. Declararea semnalelor: Acestea ar include SCL, SDA, adresa slave, și semnalele de start și stop, pentru a gestiona tranzacțiile I2C
2. Crearea unui bloc de control pentru gestionarea tranzacțiilor I2C: Acest bloc ar trebui să conțină logica pentru a genera semnalele de start și stop, pentru a trimite și primi datele de la senzorul de temperatură.
3. Definirea adresei slave: Aceasta ar trebui să fie o constantă sau variabilă care conține adresa unică a senzorului de temperatură.
4. Implementarea unui mecanism de comparare a adresei slave: Acest mecanism ar trebui să permită compararea adresei slave cu adresa senzorului de temperatură pentru a se asigura că se comunică cu dispozitivul corect.

Semnalele de start și stop sunt utilizate în protocolul I2C pentru a marca începutul și sfârșitul unei tranzacții de comunicare. Semnalul de start este generat de dispozitivul master pentru a indica începutul unei tranzacții. Acesta este generat prin coborârea nivelului semnalului SDA în timp ce nivelul semnalului SCL este ridicat. Semnalul de stop este generat de dispozitivul master pentru a indica sfârșitul unei tranzacții. Acesta este generat prin ridicarea nivelului semnalului SDA în timp ce nivelul semnalului SCL este ridicat. Logica pentru generarea semnalelor de start și stop într-un modul I2C este implementată prin utilizarea unui contor pentru a număra ciclurile de semnal SCL și prin detectarea schimbării nivelului SDA în funcție de contor. De exemplu, se poate detecta dacă SDA este coborât în timp ce SCL este ridicat, și se poate genera semnalul de start. De asemenea, se poate detecta dacă SDA este ridicat în timp ce SCL este ridicat, și se poate genera semnalul de stop.



Implementarea modului temp_sensor:

1. Inițializarea modului I2C: Se configurează semnalele SCL și SDA, precum și adresa slave a senzorului de temperatură.
2. Generarea semnalului de start: Se generează semnalul de start prin coborârea nivelului semnalului SDA în timp ce nivelul semnalului SCL este ridicat, aceasta indică începutul unei tranzacții.
3. Trimiterea adresei slave: Se trimite adresa slave a senzorului de temperatură pe semnalul SDA în timp ce semnalul SCL este ridicat.
4. Generarea semnalului de read: Se generează un semnal de citire pentru a indica că se dorește să se primească date de la senzorul de temperatură.
5. Primirea datelor: Se citesc datele de la senzorul de temperatură prin intermediul semnalului SDA în timp ce semnalul SCL este ridicat.
6. Generarea semnalului de stop: Se generează semnalul de stop prin ridicarea nivelului semnalului SDA în timp ce nivelul semnalului SCL este ridicat, aceasta indică sfârșitul unei tranzacții.
7. Procesarea datelor: După ce datele sunt primite, acestea pot fi prelucrate pentru a obține valoarea de temperatură.

Pentru implementarea modului UART au fost necesare:

1. Declararea semnalelor: Acestea ar include semnalele de intrare RX (receive) și TX (transmit), precum și semnalele de control, cum ar fi semnalul de start și cel de stop.
2. Crearea unui bloc de control pentru gestionarea tranzacțiilor UART: Acest bloc ar trebui să conțină logica pentru a genera semnalele de start și stop, pentru a detecta și a sincroniza datele de intrare și ieșire.
3. Configurarea parametrilor UART: Acestea ar include viteza de transmisie (baud rate), numărul de biți pe transmisie și configurația parității.

FSM-ul pentru protocol UART include următoarele stări și tranziții:

- Starea inițială: "Idle" - în această stare, UART-ul nu efectuează nicio tranzacție și așteaptă un semnal de start.
- Starea "Start Bit": în această stare, UART-ul detectează și sincronizează semnalul de start, care indică începutul unei tranzacții de transmisie sau recepție.
- Starea "Data Bits": în această stare, UART-ul transmite sau primește datele propriu-zise pe baza semnalelor RX sau TX.
- Starea "Stop Bit": în această stare, UART-ul detectează și sincronizează semnalul de stop, care indică sfârșitul unei tranzacții.

Tranzițiile dintre aceste stări ar putea fi definite astfel:

- Din starea "Idle" se trece la starea "Start Bit" atunci când se detectează un semnal de start.
- Din starea "Start Bit" se trece la starea "Data Bits" atunci când se sincronizează semnalul de start.
- Din starea "Data Bits" se trece la starea "Stop Bit" atunci când se transmit sau se primesc toate datele.
- Din starea "Stop Bit" se trece înapoi la starea "Idle" pentru a aștepta un nou semnal de start.

Implementare modul UCC:

- Starea inițială: "Inceput" - în această stare, UART-ul nu efectuează nicio tranzacție și așteaptă ca semnalul btn_start să fie 1.
- Starea "Incarcare Date": în această stare, se salvează datele care vor fi transmise către UART
- Starea "Octet1": în această stare, se transmit datele care trebuie transmise către UART spre semnalul de ieșire
- Starea "Transmisie1": în această stare, se transmit datele către UART și se așteaptă ca transmisia să se termine, adică semnalul "done" să fie 1
- Stările "Octet2", "Octet3", "Transmisie2" și "Transmisie3" sunt asemănătoare cu ce am prezentat mai sus, cu excepția din starea "Transmisie3" se trece în starea "Stop"
- Starea "Stop Bit": în această stare, UART-ul generează semnalul de stop, care indică sfârșitul unei tranzacții.

Implementarea convertorului din zecimal în ASCII (decToAsciiConvertor și decToAsciiUnit):

- Se verifică valoarea zecimală 0, 1, 2, ... 9
- Se returnează valoarea caracterului ASCII corespunzător 48, 49, 50, ... 58

Implementarea divizorului de frecvență:

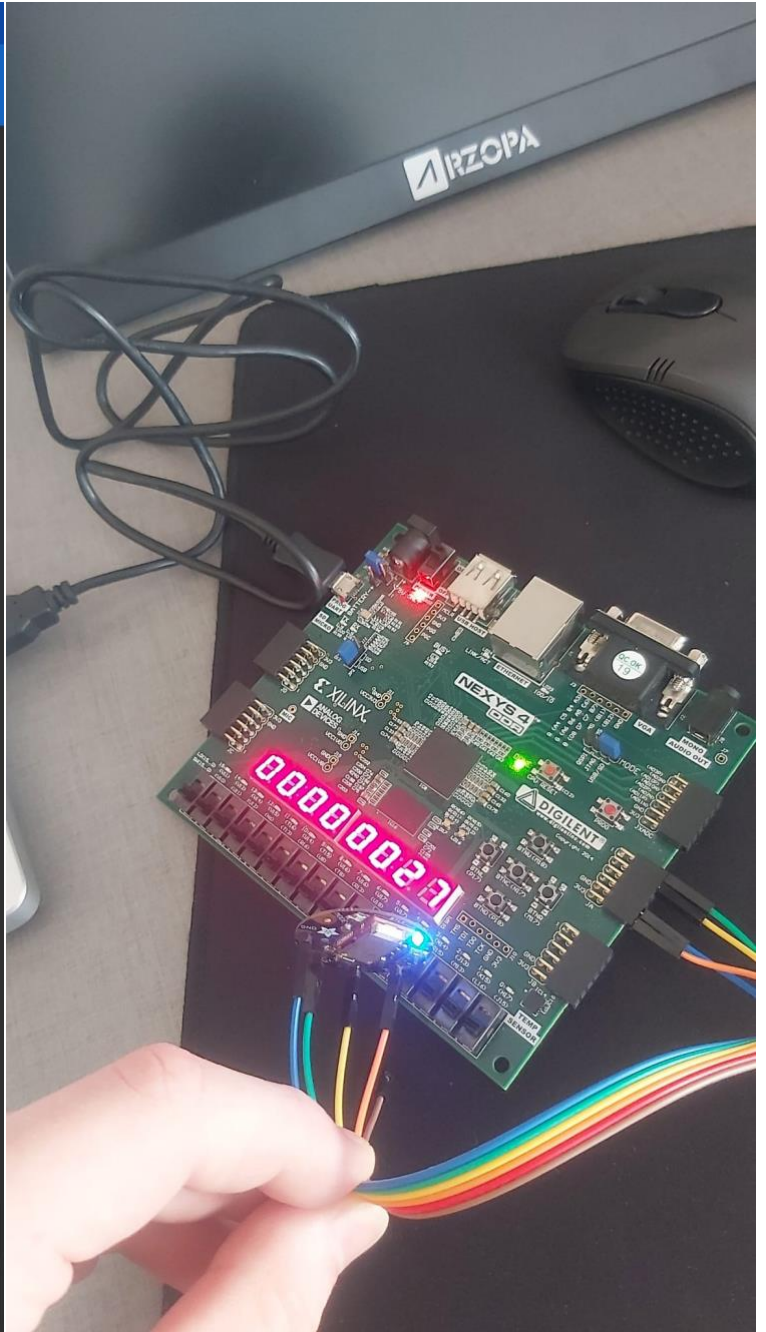
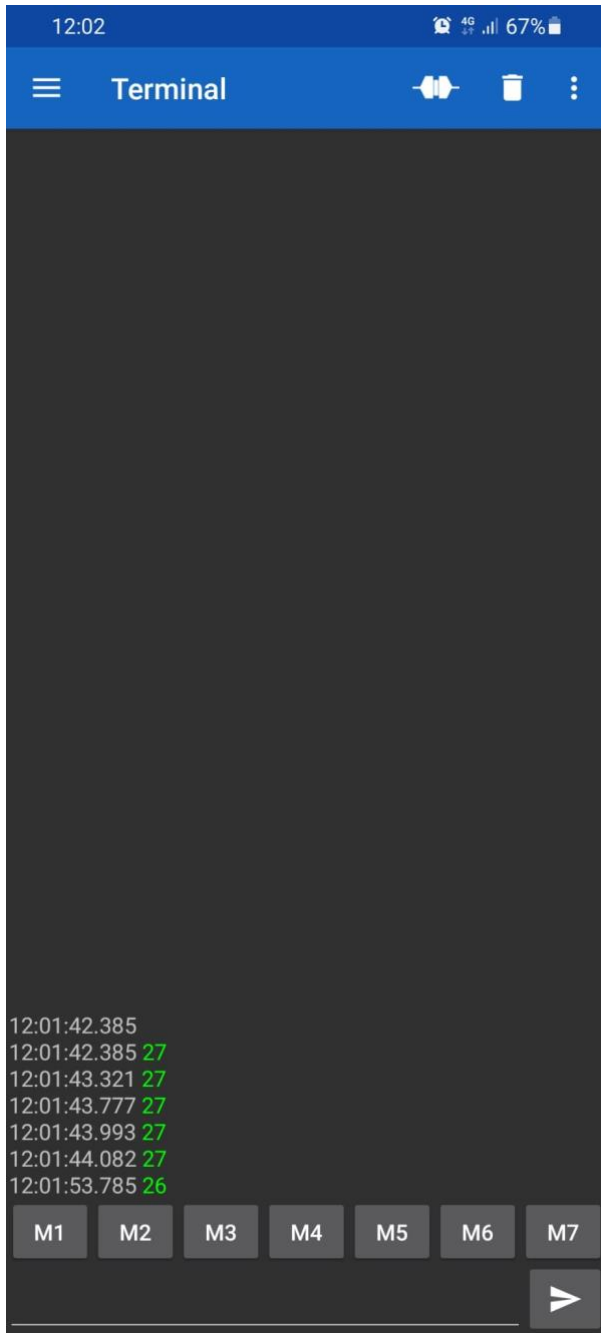
- Se numără până la 100_000
- În momentul atingerii pragului, se setează semnalul de ieșire cu valoarea 1

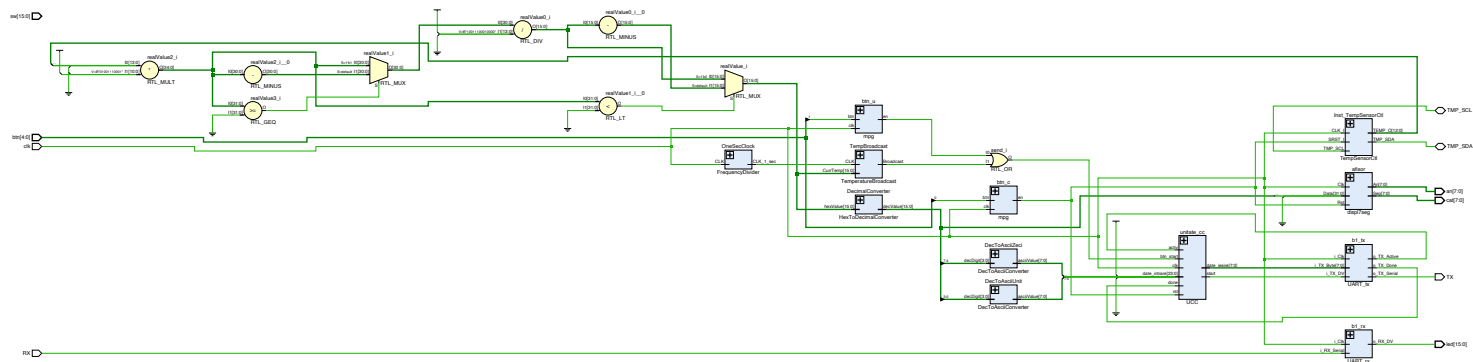
Implementarea modului temperatureBroadcast:

- În momentul primirii semnalului de clock se compară cu ultima valoare a temperaturii care a fost salvată până în momentul de față
- Dacă temperature salvată este egală cu temperature curentă semnalul broadcast rămâne 0
- Dacă temperaturile sunt diferite, semnalul broadcast devine 1 (fapt ce declanșează trimiterea automată a temperaturii către UART), după care se salvează temperature curentă

Funcționare:

- Se conectează modulul Bluetooth la placă.
- Se conectează telefonul la modul Bluetooth
- Prin apăsarea de buton se transmite temperature manual
- În momentul în care se detectează o schimbare de temperature se trimite temperatura automat
- Temperatura este primită pe telefon, în aplicația Serial Bluetooth Terminal





Concluzie

În concluzie, proiectul realizat a demonstrat capacitatea de a utiliza VHDL pentru a implementa comunicarea UART pentru citirea valorilor de temperatură de pe senzorul integrat al plăcii Nexys 4 DDR și transmiterea acestora prin Bluetooth folosind un modul Pmod BT2. Acest proiect a arătat că este posibil să se combine mai multe componente hardware și să se implementeze controlul logic necesar pentru a realiza o funcționalitate specifică.

Proiectarea hardware a fost o experiență pozitivă și interesantă, oferind oportunitatea de a înțelege mai bine funcționarea componentelor hardware și cum acestea pot fi combinate pentru a realiza o funcționalitate dorită. De asemenea, a fost o provocare să se implementeze controlul logic necesar și să se depășească problemele care au apărut în timpul proiectării.

În general, proiectarea hardware poate fi o activitate interesantă și împlinitoare, oferind posibilitatea de a crea soluții unice pentru probleme specifice. Este important să se înțeleagă bine componentele hardware și cum acestea funcționează împreună, precum și să se ia în considerare orice probleme care ar putea apărea în timpul proiectării.

Bibliografie

La baza dezvoltării proiectelor au stat și următoarele documentații și discuții ale oamenilor de specialitate:

UART:

<https://forum.digikey.com/t/uart-vhdl/12670>

<https://nandland.com/uart-rs-232-serial-port-com-port/>

<https://nandland.com/uart-serial-port-module/>

Temperature Sensor:

https://www.xilinx.com/support/documents/university/XUP%20Boards/XUPNexys4DDR/documentation/Nexys4-DDR_rm.pdf

<https://pdf1.alldatasheet.com/datasheet-pdf/view/347597/AD/ADT7420.html>