O imagine care conține text, Font, siglă, Grafică

Descriere generată automat

**Măsurarea temperaturii utilizând ADT7420**

**Studenți: Catruc Alexandru-Dan Prof. coord.: Lișman Florin**

**Gherman Cosmin-Nicolae**

**2023/2024**

Cuprins

**Rezumat**3

**Introducere**3

**Fundamentare teoretică**4

**Proiectare și implementare**5

**Rezultate experimentale**12

**Concluzii**13

**Bibliografie**13

**1. Rezumat**

Acest proiect se bazează pe măsurarea temperaturii utilizând o placă NEXYS 4 care are în componența sa un sensor ADT7420. Această valoare trebuie să fie afișată într-o aplicație Android conexiunea aplicație  <=> FPGA fiind realizată utilizând un modul Buletooth HC05. Totuși această valoare poate fi văzută și pe SSD-ul plăcii pentru a se vedea corectitudinea rezultatelor afișate pe telefon.

Cerinte funcționale :

1. Dezvoltarea unei probleme de hardware utilizând limbajul VHDL
2. Utilizarea plăcuței Nexys 4
3. Folosirea senzorului incorporat pe placă ADT7420
4. Conexiune cu Bluetooth
5. Aplicație pe telefon

Acest proiect a fost unul interesant și a reușit să ne mobilizeze să învățăm lucruri noi despre lucratul cu VHDL. Ajutați de documentațiile găsite pe internet și de timpul petrecut analizării tematicii, s-a reușit finalizarea cu succes a cerințelor. Temperatura a fost transmisă corect și continuu pe telefon în aplicație.

**2. Introducere**

Un prim start a fost inițiat de căutarea de informații și componente necesare. Am realizat necesitatea unui modul I2C pentru transmiterea de informații de la senzorul de temperatura la placa. Implementarea unui modul I2C cât și a citirii unei valori de temperatură a fost influențată și de forum-urile de pe Internet astfel incât sa reușim o modificare a temperaturii senzorului aproape instantă. Pentru a verifica corectitudinea am apelat la un afișor SSD. Inițial acesta era in format hexazecimal și era destul de dificilă o verificare clară a temperaturii. După mai multe testări, s-a concluzionat faptul că temperatura de pe senzor este corectă și se poate trece la următorul pas.

Pentru partea de Bluetooth am utilizat senzorul HC05 care are urmatorii pini: VCC, GND, TXD, RXD. Alimentarea cu energie electrică se realizează prin pinii de VCC (5V) și GND (0V) iar transmisia se face prin TXD (Transmit Data) si RXD (Recive Data). Acestă sensor va primi informatie de la plăcuță și o le va trimite constant mai departe către aplicatia Android.

Partea de cod pentru crearea unei aplicatii s-a dezvoltat in AndroidStudio. Aplicația Android se bazează pe utilizarea a 4 meniuri: meniul de start, meniul pentru temperatura în grade Celsius și Fahrenheit și cel din urmă meniul de conectare cu Bluetooth. Navigarea intre aceste interfețe este independentă și foarte inteligibilă pentru utilizatori.

Utlima parte a acestui proces a fost consemntată de lucrul în timpul laboratorului, unde am creat un UART (conexiune pe baza recive/transmit). Prin ajutorul aplicației PUTTY am verificat transmisiunea de temperatura de la placă către laptop. Ulterior după ce s-a reușit acest prag, proeictul a fost aproape terminat, trecerea de la PUTTY la aplicație fiind aproximativ la fel. S-a convertit valoarea temperaturii astfel încât să fie afișată în decimal și aceasta valoare a fost pusă atât pe SSD cât și pe telefon într-un ciclu continuu.

**3. Fundamentare teoretică**

UART, cunoscut și ca Universal Asynchronous Receiver Transmitter, este un dispozitiv de circuit integrat ce facilitează schimbul de date între două entități prin intermediul unei conexiuni seriale. Acesta transformă semnalele digitale în semnale analogice pentru transmiterea pe o linie serială și viceversa, convertind semnalele analogice înapoi în format digital pentru procesare de către sistem.

Bluetooth UART reprezintă o variantă a protocolului de comunicație UART, adaptată pentru schimbul de date între dispozitive Bluetooth. Acest protocol funcționează similar cu UART tradițional, dar diferă prin faptul că transmite datele printr-o conexiune wireless Bluetooth, în locul unei linii seriale fizice. În contextul Bluetooth UART, transferul de date se efectuează prin intermediul unui pin RX (receptor) și unui pin TX (transmițător) pe modulul Bluetooth. Pinul RX este folosit pentru a recepționa datele de la dispozitivul conectat, iar pinul TX pentru a trimite date către acesta. Când un dispozitiv trimite date către un altul, informațiile sunt convertite în semnale digitale și expediate prin pinul TX. Acestea sunt apoi primite de pinul RX al dispozitivului receptor, unde sunt convertite înapoi în semnale digitale pentru procesare.

Senzorul de temperatură utilizat în proiectul nostru este ADT7420, un senzor cu o mare precizie și un timp de răspuns rapid. Acesta poate măsura temperaturi între -40°C și +125°C și oferă o precizie de ±0.5°C în intervalul -40°C până la +85°C, și de ±1°C între -40°C și +125°C. Pentru a citi temperaturile de la senzorul ADT7420, se va folosi interfața I2C sau SPI pentru a comunica cu senzorul și pentru a extrage valori de temperatură. Acest proces implică trimiterea de comenzi către senzor și citirea datelor, care apoi sunt procesate și afișate.

Protocolul Bluetooth HC-05 este un modul versatil și accesibil, folosit adesea în proiectele de electronice pentru a adăuga funcționalitate Bluetooth. HC-05 poate funcționa atât ca un master cât și ca un slave, facilitând comunicarea wireless între dispozitive. Modulul permite comunicația serială prin Bluetooth, având un raza de acoperire de aproximativ 10 metri în condiții ideale.

Android Studio, pe de altă parte, este un mediu de dezvoltare integrat (IDE) pentru dezvoltarea aplicațiilor Android. Acesta oferă o gamă largă de instrumente pentru codare, debug și testare, fiind o alegere populară pentru dezvoltatori

Interfața I2C (Inter-Integrated Circuit), este un protocol de comunicație multi-master, multi-slave, utilizat pe scară largă pentru conectarea microcontrolerelor și a diferitelor dispozitive periferice, cum ar fi senzori și ecrane. Aceasta permite comunicația bidirecțională folosind doar două fire: SCL (Serial Clock) și SDA (Serial Data). Este apreciată pentru simplitatea sa și pentru capacitatea de a conecta multiple dispozitive.

**4. Proiectare și implementare**

4.1. Etape fundamentale pentru implementarea hardware :

1. Prelucrarea de informații primite de la senzorul ADT7420
2. Afișarea pe SSD a informațiilor (temperature)
3. Modificarea din hexazecimal în decimal a temperaturii de pe SSD
4. Utilizarea unui UART pe 8 biți pentru o transmitere parțială de temperatură
5. Testarea transmiterii parțiale cu PUTTY
6. Maparea UART 8 biți în UART 16 biți pentru transmiterea intregii temperaturi
7. Testarea finala cu PUTTY pentru a vedea temperatura
8. Convertirea din decimal în Ascii pentru transmisia la UART 16
9. Testarea finala cu PUTTY pentru a vedea temperatura în decimal
10. Adăugarea modulului Bluetooth și conexiunea acestuia cu placa
11. Realizarea aplicatiei Android
12. Trimiterea datelor finale (temperaturii) către telefon
13. Conexiunea Bluetooth modul HC-05 și telefon
14. Primirea datelor și afișarea lor

4.1.1. Utilizând TechForum-ul de la DigiKey[[1]](#footnote-1) am conectat I2C ul și senzorul de temperatură la proiectul nostru. Pentru I2C inițial s-au declarat semnalele de intrare și iesire în entitate. Aici se specifică parametrii generici ai modulului cum ar fi: frecvența ceasului de intrare și frecvența bus-ului. În portul modulului se regăsesc semnale de control(ena), adresă (addr), de citire si scriere (vectori de biti) și SDA (Serial Data Line), SCL (Serial Clock Line) acestea fiind in-out-uri. Primul proces din arhitectura are rolul de a genera semnalele de ceas (scl\_clk) și date (data\_clk) pentru controlul sincronizării utilizând variabila count pentru a numara ciclurile de ceas necesare si a genera semnale corespunzătoare. Tot acest proces trebuie să fie în conformitate cu protocolul I2C ținând cont de evenimentele de pe semnalele de clock si reset. Urmotorul proces are la baza crearea unui state machine pentru a controla fiecare etapă. Ca prim lucru se inițializează starea si variabilele asociate in cazul in care resetul este activat ulterior în funcție de starea mașinii se reazlizează diferite actiuni. Acestea sunt clasate în; gestionarea comenzilor de citire/scriere (command, rd, wr), și oprirea tranzacției (stop). Următoarea etapă constă în folosirea I2C-ului ca o componentă a senzorului de temperatură. Se definește entitatea modulului temp\_senzor cu parametri generici pentru frecvența ceasului și adresa I2C a senzorului. Porturile includ semnalele de ceas (clk), reset (rst), scl și sda pentru linia I2C, semnalul de eroare de acknowledge I2C (i2c\_ack\_err), și semnalul de temperatură (temperature). Se instanțiază modulul I2C folosind parametrii specificați în generic map și port map. Și în cadrul acestei componente se va folosi un state machine care va fi constituit din următoarele stări :

* "start" - în acest caz, se numără ciclurile de ceas pentru a aștepta un interval de timp (1/10 dintr-o secundă) înainte de a trece la următoarea stare (init)
* "init" - verifică dacă modulul I2C este ocupat (i2c\_busy). În cazul în care a fost ocupat și acum este disponibil, se trece la următoarea stare (pause). Se gestionează un contor (busyCount) pentru a executa anumite acțiuni în mod repetat în diferite cicluri de ceas
* "pause" - se așteaptă un interval de timp înainte de a trece la următoarea stare (readData)
* "readData" - similar cu cazul "init", verifică starea de ocupare a modulului I2C și realizează acțiuni specifice pentru citirea datelor de temperatură
* "outputRes" - stabilește datele de temperatură (temperature) pe baza datelor citite de la senzor și se pregătește pentru următoarea stare (pause).

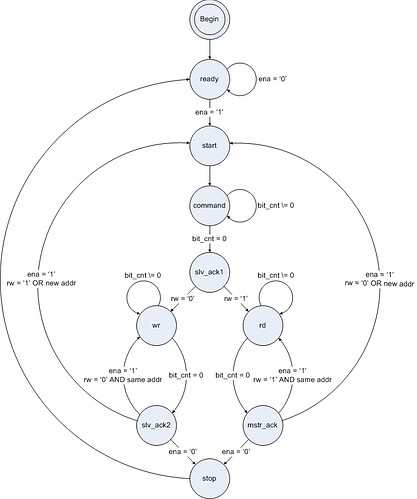
Acesta este o descriere a funcționalității și o prezentare generală pe baza codului. Acesta implementează o mașină de stări pentru a controla tranzacțiile I2C și pentru a citi datele de temperatură de la senzor într-un mod sincronizat cu semnalul de ceas.

Temp State Machine:

O imagine care conține cerc, diagramă, linie, captură de ecran

Descriere generată automat

I2C State Machine :



4.1.2. Codul pentru SSD este preluat din anii trecuți, unde esenta este selectia pentru a ști cum afișăm temperatura corect. În funcție de acestă selecție anodul va primi o anumita valoare corespunzătoare cu poziția de afișat pe SSD. Pentru catod lucrurile sunt simple, o să punem hardcodat ^C și o să afisam si un punct după primele 2 cifre pentru a face diferența între partea întreaga și cea zecimală.

4.1.3. Aici lucrurile sunt simple, cu ajutorul librariei ARITH.ALL și UNSIGNED.ALL am folosit funcția conv\_integer care converteste o valoare din binar in una de tip integer. După aceasta convertire în integer am preluat fiecare cifră din numar și s-a stocat in vectori diferiți pe 4 biți. Pe baza unui convertor decimal to Ascii, s-a putut afișa pe ecran o valoare decimal a temperaturii.

4.1.4 Acest modul implementează o unitate de transmitere UART, care poate fi folosită pentru a trimite date pe o linie serială. Transmiterea se realizează asincron, iar viteza de transmisie poate fi specificată prin parametrul generic n. Entitatea UART\_tx are un port de intrare pentru date (TxData), semnalele de ceas (Clk), reset (Rst), semnalul de start (Start) și un port de ieșire pentru semnalele de transmisie (Tx) și semnalul de gata pentru transmitere (TxRdy). Se definește o constantă n pentru viteza de transmisie în bps (biți pe secundă). Restul de cod este preluat din laborator[[2]](#footnote-2).

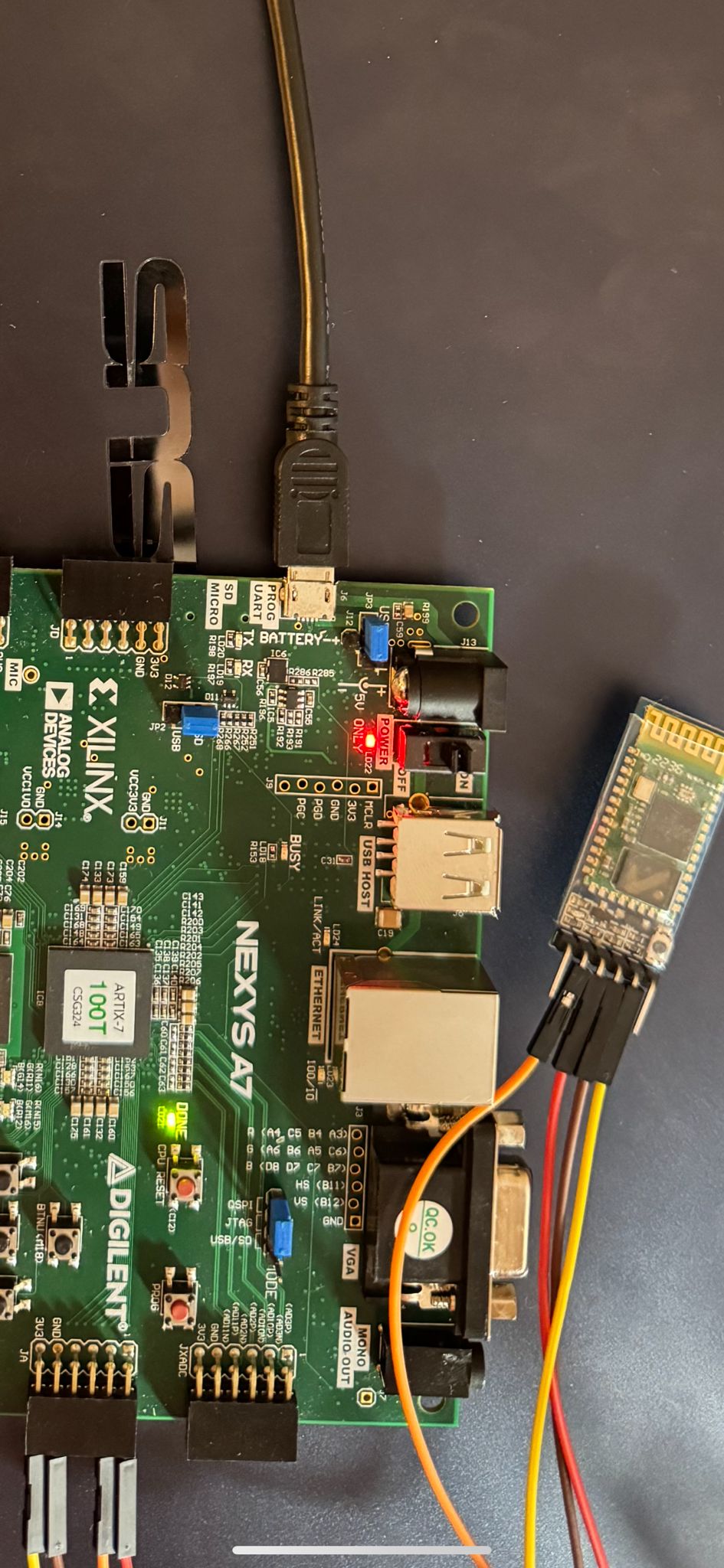
O imagine care conține cerc, diagramă, artă

Descriere generată automat

4.1.5. Testarea cu PUTTY a fost foarte utilă din cauza impedimentelor pe care le am intâmpinat cu modulele Bluetooth. Astfel, cu ajutorul laboratorului2, am reușit să dezvoltăm module de simulare live.

4.1.6. Reușind să transmitem o porțiune din temperatură pe 8 biți am creată o componentă nouă pe baza căreia putem să transmitem toti cei 16 biți de temperature utilizând transmisia pe 8 prin mapare. Totul se ghideaza in jurul vectorului de biți temp care primește valoarea corectă în funcție de nr\_octeti. Dacă nr\_octeti este 7, se iau primele 8 biți, iar apoi se deplasează în funcție de nr\_octeti.

4.1.10.



4.1.11. Descrierea aplicatiei Android. Aplicația utilizată pentru afișarea temperaturii a fost dezvoltată cu ajutorul IDE – ului IntelliJ IDEA folosind plugin- ul pentru Android. Limbajul de programare folosit a fost Java pentru partea de Back – End ( Logic/ Application Layer), în timp ce pentru Front – End ( UI Layer ) s-a folosit XML. Interfața cu utilizatorul este una prietenoasă, constând în 4 meniuri principale:

1. Meniul de start – este cel care apare când aplicația este deschisă, utilizatorul având posibilitatea de a face 3 acțiuni principale: vizualizarea temperaturii în grade Celsius, vizualizarea temperaturii în grade Fagrenheit, conectarea la un dispozitiv Bluetooth cu care dispozitivul Android este asociat.

2. Meniul pentru temperatura în grade Celsius – este cel care afișează temperatura în grade Celsius în timp real cât timp aplicația Android este conectată la dispozitivul Bluetooth. În caz contrar, cât timp nu există conexiune, nu se va afișa nimic.

3. Meniul pentru temperatura în grade Fahrenheit – este la fel ca și cel pentru temperatura în grade Celsius, dar afișeaza temperatura în grade Fahrenheit.

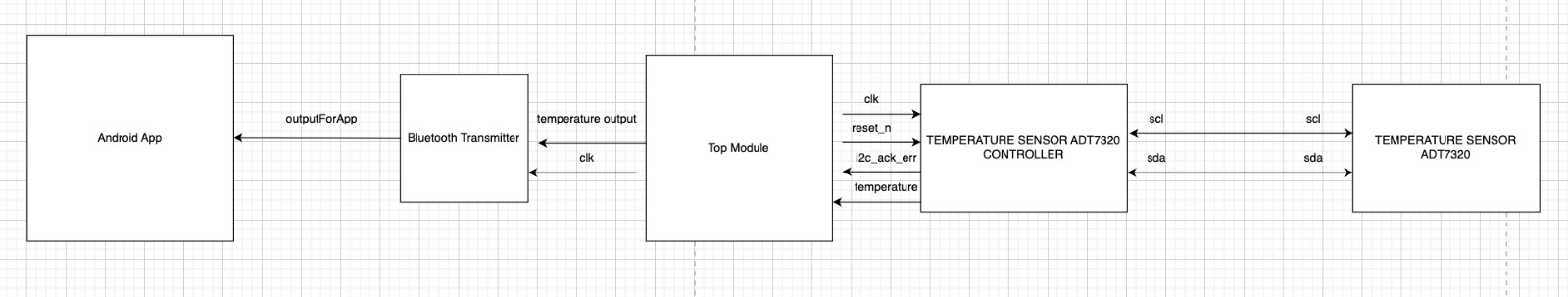
4. Meniul pentru conexiunea Bluetooth – ajută la conectarea aplicației Android la un dispozitiv Bluetooth asociat. Utilizatorul va alege dintr-o listă dispozitivul dorit și va face click pe cel la care dorește să se conecteze.

Schemă finală :

O imagine care conține text, diagramă, Plan, Desen tehnic

Descriere generată automat

Schemă bloc :

**5. Rezultate experimentale**

Se poate observa în imagine aplicația pe telefon, modulul Bluetooth și plăcuța Nexys4 cu afișajul SSD al temperaturii. Există un mic delay intre afișarea temperaturii pe telefon și cea de pe placă din cauza modulului. Exemplu :

**O imagine care conține electronice, Inginerie electronică, text, Dispozitiv electronic

Descriere generată automat**

**6. Concluzii**

În concluzie, proiectul a fost realizat cu success, iar impedimentele de pe parcurs ne-au făcut să progresăm în ceea ce înseamnă programarea în limbajul VHDL. Îmbinarea componentelor de UART si I2C în proiect a constituit partea dificilă, din care cred că am rămas cu câteva noțiuni despre ce înseamnă comunicare Bluetooth și cea serială.

Pe baza acestuia s-au dezvoltat si skill-uri noi de Java utilizând pentru prima dată AndroidStudio, realizând conexiunea cu placa NEXYS 4.

Toate aceste concepte au condus la o muncă în echipă fară de care rezultatul, cu siguranță, nu ar fi fost cel așteptat.

**7.** **Bibliografie**

<https://forum.digikey.com/t/temperature-sensor-adt7420-pmod-controller-vhdl/20296>

<https://forum.digikey.com/t/i2c-master-vhdl/12797>

<https://www.youtube.com/watch?v=4W9MQa3jBm8>

L8\_9 - Testare-Depanare.pdf >> laborator

<https://youtu.be/iFtjox9_zAI?si=QFLz3aL3XWcc3ezx>

ADT7420 datasheet

1. https://forum.digikey.com/t/temperature-sensor-adt7420-pmod-controller-vhdl/20296 [↑](#footnote-ref-1)
2. L8\_9 - Testare-Depanare.pdf [↑](#footnote-ref-2)