



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

# Masurarea temperaturii cu senzorul de pe placa Nexys 4 DDR si transmiterea ei la un dispozitiv mobil

**Studenti: Dorofte Andrei**

**Iacob Liviu-Mihai**

**Profesor indrumator: Lisman Dragos Florin**

**Grupa: 302310**



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

## Cuprins

1. Rezumat.....	3
2. Introducere.....	4
2.1 Comunicarea serială.....	4
2.2 Interfața I <sup>2</sup> C.....	5
2.3 Interfața UART.....	6
2.4 Modulul Bluetooth Pmod BT2.....	7
2.5 Senzorul de temperatură ADT7420.....	8
3. Fundamentare teoretică.....	9
4. Proiectare și implementare.....	11
5. Rezultate experimentale.....	20
6. Concluzii.....	22
7. Bibliografie.....	23



# UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

## 1. Rezumat

Proiectul are ca scop masurarea temperaturii ambientale cu ajutorul senzorului de temperatura de pe placa FPGA Nexys4 DDR. Ulterior, aceasta temperatura trebuie transmisa unui dispozitiv mobil, conectat la placa printr bluetooth. Pentru aceasta conexiune se foloseste modulul bluetooth PmodBT2, iar informatia este trimisa catre acesta printr-o interfata de comunicare UART.

Pe de alta parte, datele receptionate de la senzorul de temperatura sunt trimise printr-o interfata I2C catre placa FPGA. Pentru implementare a fost folosit limbajul de descriere hardware VHDL, utilizand Vivado 2020.1 pentru simulare si testare.

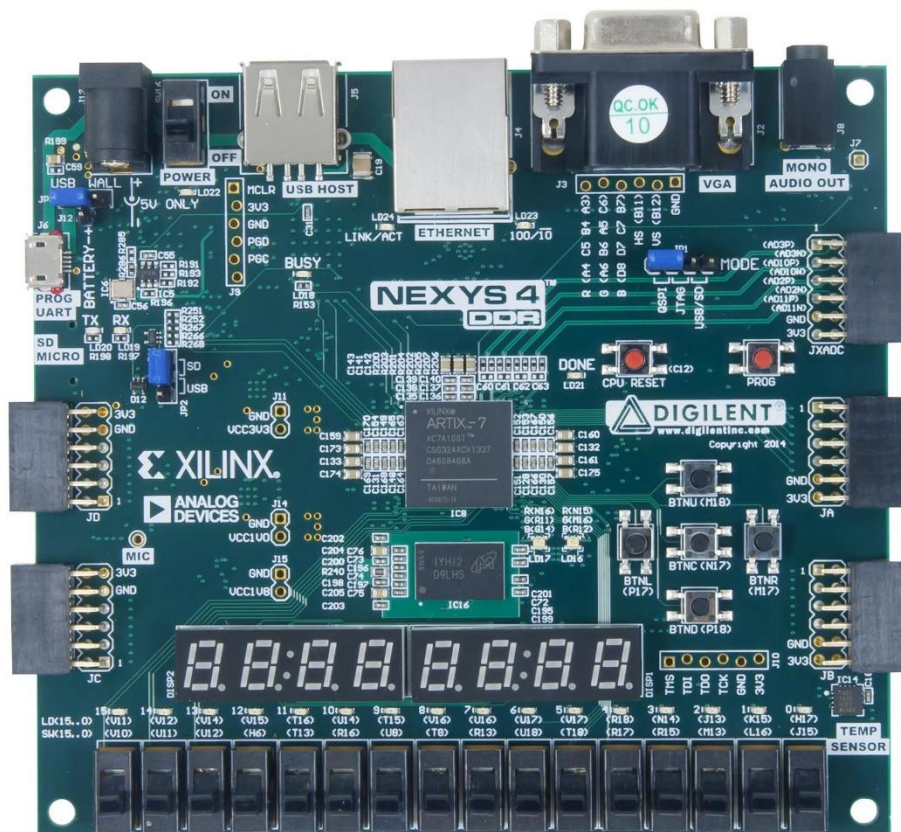


Figura 1

– Placa de dezvoltare Nexys4 DDR



## 2. Introducere

Tema proiectului este realizarea comunicatiei dintre 2 dispozitive, folosind interfata bluetooth. Cele 2 dispozitive, in cazul nostru, au fost o placa de dezvoltare Nexys4DDR si un telefon mobil cu Android. Componentele folosite pentru realizarea comunicarii au fost senzorul de temperatura integrat al placii si modulul PmodBT2, pentru realizarea conexiunii cu telefonul mobil.

### 2.1 Comunicare seriala

Transmisia de date seriala, este o tehnologie de comunicare intre calculator si dispozitivele periferice, in care bitii de date sunt transferati succesiv de-a lungul unui canal de comunicare.

In modul sincron, emitatorul si receptorul au acelasi semnal de tact. Caracterele sunt transmise rapid, unul dupa altul, fara biti de start si stop. Pentru sincronizare, mesajul transmis este precedat de caractere speciale de sincronizare, detectabile de circuitul receptorului. Aceasta modalitate de comunicare este de multe ori mai rapidă, cu toate acestea are nevoie de cel puțin un fir in plus, pentru transmiterea semnalului de ceas.

In transferul asincron, emitatorul si receptorul au semnale de tact separate, sincronizarea la nivel de bit este asigurata numai pe durata transmisiei efective a fiecarui caracter. O asemenea comunicare este orientata pe caractere individuale si are dezavantajul ca necesita informatii suplimentare in proportie de cel puțin 25% pentru identificarea fiecarui caracter si o atenție sporita trebuie acordata sincronizării datelor transferate.

Din punct de vedere al canalelor de comunicare, interfetele de comunicare seriala se impart in: Simplex, Half-Duplex si Full-Duplex. Aceste canale ofera cai de transmitere a informatiilor si datorita lor, informatia este transmisa fara obstructie.

- Simplex necesita un singur fir de transmisie, iar datele se pot transfera intr-o singura directie;
- Half-duplex functioneaza alternative ca transmitter, iar apoi ca receiver necesitand de asemenea de un singur fir de transmisie;
- Full-duplex transfera date simultan in ambele directii, avand nevoie de doua fire de transmisie.



## 2.2 Interfata I2C

Interfata de comunicație I2C permite unui circuit integrat „master”, să comunice cu unul sau mai multe dispozitive „slave”. I2C necesită doar 2 fire de comunicație între master și slave, iar un dispozitiv master poate suporta până la 1008 dispozitive slave. De asemenea, I2C suporta un mod multi-master, prin care mai multe dispozitive master pot comunica cu toate dispozitivele de pe magistrală.

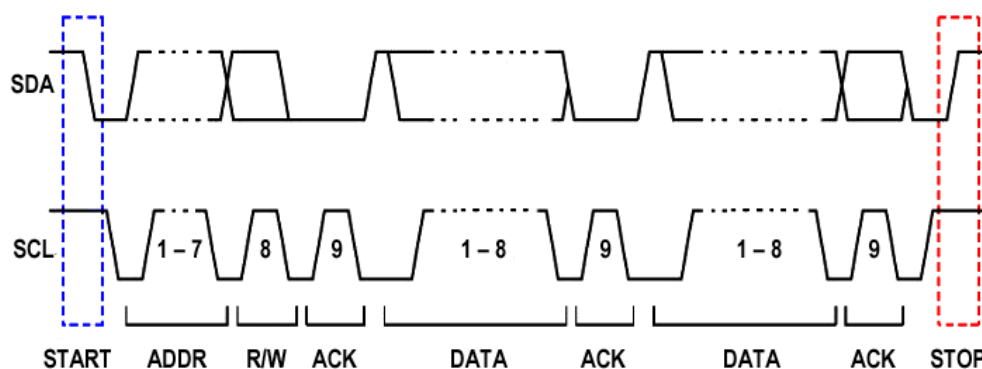


Figura 2 – modul de transmisie I2C

Sincronizarea datelor este asigurată de cele 2 semnale SDA și SCL, ambele bidirectionale. Astfel, o transmisie se începe cu un bit de start și se termină printr-un bit de stop. Condiția de start este trecerea liniei SDA de la 1 la 0, SCL rămânând în continuare 1, iar condiția de stop este reprezentată de trecerea liniei SDA de la 0 la 1, SCL fiind 1, asemenea condiției de start. În continuare, la fiecare impuls de ceas se transmite un bit de date, iar linia SDA se modifică doar atunci când linia SCL este pe 0 logic, iar ceasul sistemului pe 1 logic. Datele se transmit în câte 7 biți, urmate de un bit ack, care semnalează faptul că datele au fost transmise sau recepționate cu succes.

## 2.3 Interfata UART

UART (Universal Asynchronous Receiver/Transmitter) este o interfata seriala asincrona care nu transmite semnal de ceas prin linia de date seriala. Receptorul recunoaste valorile binare individuale fara o linie comuna de ceas. Acesta poate funcționa folosind 3 fire: GND, Transmit (Tx) și Receive (Rx).

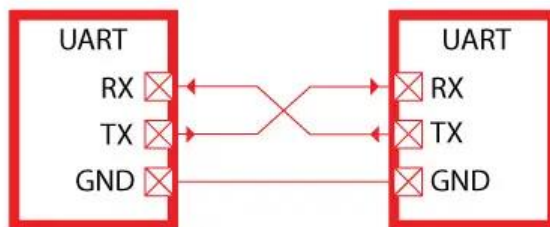


Figura 3 – Interfata UART

Interfata UART consta din doua parti:

- un receptor (receiver - Rx) care convertește un flux serial de biți în date paralele (cuvinte) pentru microprocesor;
- un transmitator (transmitter - Tx) care convertește date paralele de la microprocesor într-un flux serial de biți, pentru transmisie.

UART-urile au mai multi parametri care pot fi setați de către utilizator. Aceștia sunt:

- Rata de transfer (baud rate) – 9600, 19200, 115200(reprezentand biți pe secundă) - viteza la care sunt transmise datele seriale.
- Numărul de biți de date
- Bitul de paritate - Un bit de paritate poate fi adăugat după trimiterea datelor(optional). Paritatea este întotdeauna calculată prin efectuarea unei operații XOR pe toți biți transmisi.



## UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

- Bitii de sincronizare –bitii de START si de STOP, marcând începutul și sfârșitul unui pachet de date. Un bit de oprire este întotdeauna setat la 1 și pot fi 1 sau 2 biți de stop, în vreme ce bitul de pornire este doar unul și este setat la 0.
- Fluxul de control - Controlul fluxului nu este utilizat în mod obișnuit în aplicațiile actuale și va fi setat la „none”.

### 2.4 Modulul bluetooth PmodBT2

Modulul PmodBT2 este un dispozitiv periferic, care foloseste modulul radio RN-42, care se conecteaza la placa FPGA Nexys4 DDR, printr-un port de 12 pini, si comunica cu aceasta, prin protocolul UART, folosind 8 biti de date, un bit de stop si fara bit de paritate. Se poate comunica si prin protocolul SPI, daca se updateaza firmware-ul RN-42.



Figura 4 – modulul PmodBT2

Modulul este compatibil cu versiunile de Bluetooth 2.1/2.0/1.2/1.0, si are 4 pini suplimentari, de tip jumper, care pot configura diferite setari ale modului:

- JP1 (PIO4) - Factory Default
- JP2 (PIO3) - Auto Discovery/Pairing
- JP3 (PIO6) - Auto Connect
- JP4 (PIO7) - Baud Rate Setting: 9600 – default is 115200





**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

## 2.5 Senzorul de temperatura ADT7420

Senzorul de temperatura integrat al placii Nexys4 DDR ofera o acuratete de 0.0078 grade Celsius, atunci cand se furnizeaza temperatura in formatul de 16 biti, si o acuratete de 0.0625 grade Celsius, atunci cand se alege formatul de 13 biti.

### TEMPERATURE CONVERSION FORMULAS

#### 16-Bit Temperature Data Format

$$\text{Positive Temperature} = \text{ADC Code (dec)} / 128$$

$$\text{Negative Temperature} = (\text{ADC Code (dec)} - 65,536) / 128$$

where ADC Code uses all 16 bits of the data byte, including the sign bit.

$$\text{Negative Temperature} = (\text{ADC Code (dec)} - 32,768) / 128$$

where Bit 15 (sign bit) is removed from the ADC code.

#### 13-Bit Temperature Data Format

$$\text{Positive Temperature} = \text{ADC Code (dec)} / 16$$

$$\text{Negative Temperature} = (\text{ADC Code (dec)} - 8192) / 16$$

where ADC Code uses the first 13 MSBs of the data byte, including the sign bit.

$$\text{Negative Temperature} = (\text{ADC Code (dec)} - 4096) / 16$$

where Bit 15 (sign bit) is removed from the ADC code.

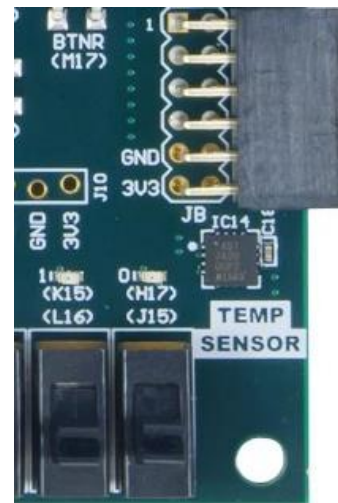


Figura 5 – ADT7420

Figura 6 – ADT7420 data format

De asemenea, senzorul de temperatura contine si registre, din care amintim: Temp MSB(adresa 0x00), Temp LSB(adresa 0x01) si registrul de configurare, cu adresa 0x03, care trebuie setat mai intai de toate, in cazul nostru cu valoarea x"00", pentru formatul ales.

Bit	Default Value	Type	Name	Description
[1:0]	00	R/W	Fault queue	These two bits set the number of undertemperature/overtemperature faults that can occur before setting the INT and CT pins. This helps to avoid false triggering due to temperature noise. 00 = 1 fault (default). 01 = 2 faults. 10 = 3 faults. 11 = 4 faults.
2	0	R/W	CT pin polarity	This bit selects the output polarity of the CT pin. 0 = active low. 1 = active high.
3	0	R/W	INT pin polarity	This bit selects the output polarity of the INT pin. 0 = active low. 1 = active high.
4	0	R/W	INT/CT mode	This bit selects between comparator mode and interrupt mode. 0 = interrupt mode 1 = comparator mode
[6:5]	00	R/W	Operation mode	These two bits set the operational mode for the <a href="#">ADT7420</a> . 00 = continuous conversion (default). When one conversion is finished, the <a href="#">ADT7420</a> starts another. 01 = one shot. Conversion time is typically 240 ms. 10 = 1 SPS mode. Conversion time is typically 60 ms. This operational mode reduces the average current consumption. 11 = shutdown. All circuitry except interface circuitry is powered down.
7	0	R/W	Resolution	This bit sets up the resolution of the ADC when converting. 0 = 13-bit resolution. Sign bit + 12 bits gives a temperature resolution of 0.0625°C. 1 = 16-bit resolution. Sign bit + 15 bits gives a temperature resolution of 0.0078°C.





### 3. Fundamentare teoretică

#### 3.1 Automate cu stări finite

Un automat cu stări finite este un circuit secvențial cu un număr limitat de stări. Aceste stări sunt folosite pentru a modela funcționalitatea automatului. Sarcina unui FSM este de a trece dintr-o stare în alta și de a seta semnalele de control necesare pentru starea următoare.

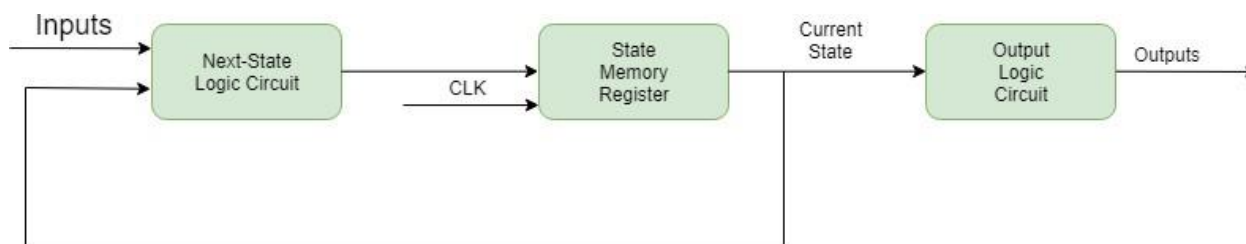


Figura 7 – automate de stare

#### 3.2 Protocolul I2C

Datele în cadrul acestui protocol sunt transmise ca mesaje. În aceste mesaje, pe lângă datele care se doresc să fie transmise, se transmit și anumiți biți de control.

Condiția de start: linia SDA trece de la un nivel înalt (1 logic) la un nivel scăzut (0 logic), înainte ca SCL să treacă de la 1 la 0. Linia de date este considerată ocupată până la primirea condiției de start.

După condiția de start, o adresă a unui Slave (de 7 biți) este trimisă pe magistrală către Master, urmată de încă un bit care reprezintă operația care se dorește să se efectueze asupra datelor: 0 pentru scriere, 1 pentru citire.

Condiția de stop: linia SDA trece la 0 la 1 după ce linia SCL trece de la 0 la 1.

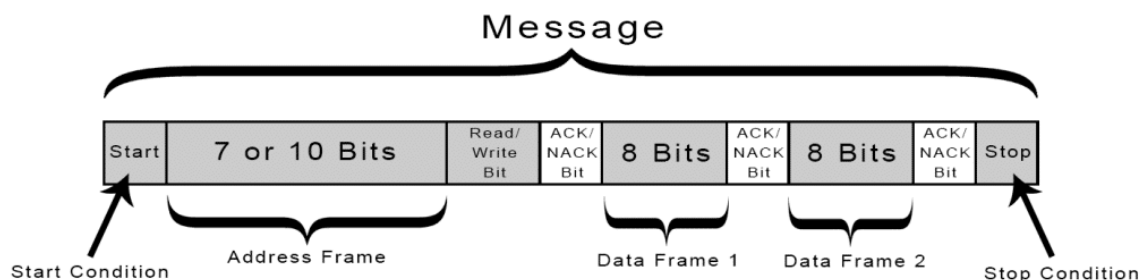


Figura 8 – Pachet de date I2C



### 3.3 Protocolul UART

Funcționalitatea acestui protocol se bazează pe faptul că transmitatorul primește informația de pe magistrală într-o formă paralelă. După primirea datelor, acesta adaugă bitul de start, de paritate și de stop, creând astfel un pachet de date, apoi pachetul este transmis serial, bit cu bit.

Receptorul primește datele serial, înlătură bitii adăugați de transmitator și convertește datele în formă paralelă.

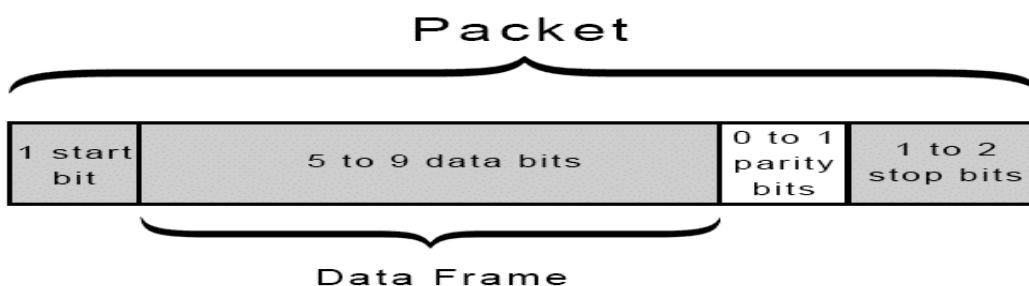


Figura 9 – pachet de date UART

### 3.4 Senzorul de Temperatura

Placa Nexys4 DDR conține un senzor de temperatură integrat (ADT7420). Senzorul are 2 configurații, una pe 13 biți și cealaltă pe 16. Configurația pe 16 biți este mai exactă, având o acuratețe de 0.025 grade Celsius. Comunicarea dintre acest senzor și placă se face prin intermediul protocolului I2C.

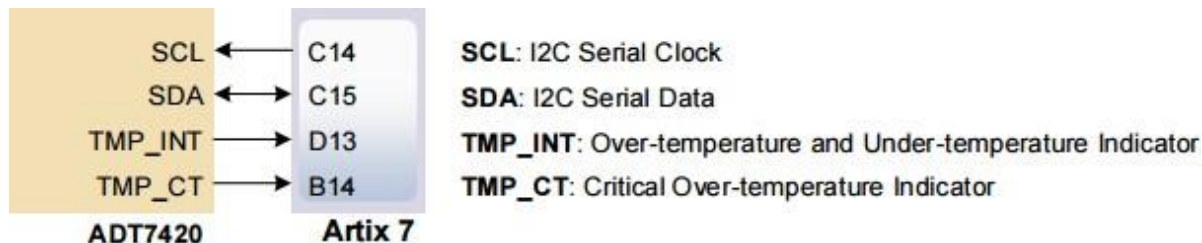


Figura 10 – pini senzor de temperatură



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

### 3.5 Soluția aleasă

Problema pe care am ales să o implementăm se referă la citirea temperaturii cu ajutorul senzorului ADT7420 de pe placa Nexys4 DDR. Citirea se realizează prin interfața I2C, iar apoi este transmisă prin interfața UART către modulul Bluetooth. De aici, informația este trimisă către un dispozitiv mobil, unde se poate vizualiza prin intermediul aplicației Android.

## 4. Proiectare și implementare

### 4.1 Arhitectura sistemului

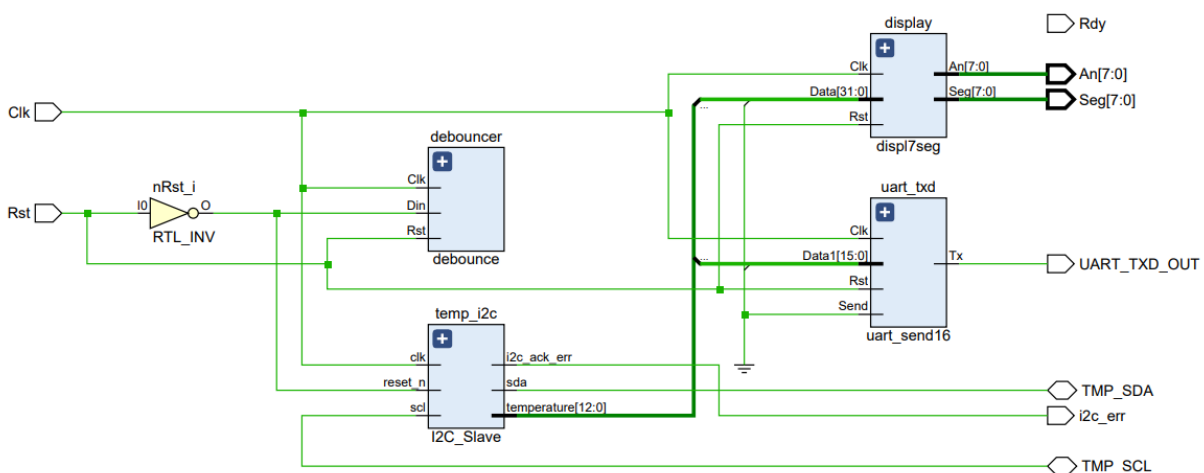


Figura 11 – RTL Schematic

Arhitectura sistemului este compusă dintr-un modul care reprezintă senzorul - I2C Slave și un modul pentru afișarea temperaturii trimise pe display, pentru a verifica corectitudinea datelor recepționate pe aplicația Android.



**UNIVERSITATEA TEHNICĂ**  
**DIN CLUJ-NAPOCA**  
**FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE**

Modulul I2C\_Slave contine o instanta I2C Controller si permite citirea temperaturii printr-un transfer de date de tip master-slave.

Intrari:

- clk => clock-ul sistemului
- rst => resetare prin apasarea butonului de pe placa

Iesiri:

- An (7:0) si Seg(7:0) – anozii si segmentii display-ului
- I2c\_err – eroare de transmisie sau receptie pe i2c, afisata pe un led
- UART\_TXD\_OUT – iesirea modului de transmisie UART
- TMP\_SDA si TMP\_SCL de tip inout, pentru conexiunea cu senzorul

## 4.2 Modulele sistemului

### I2C Controller

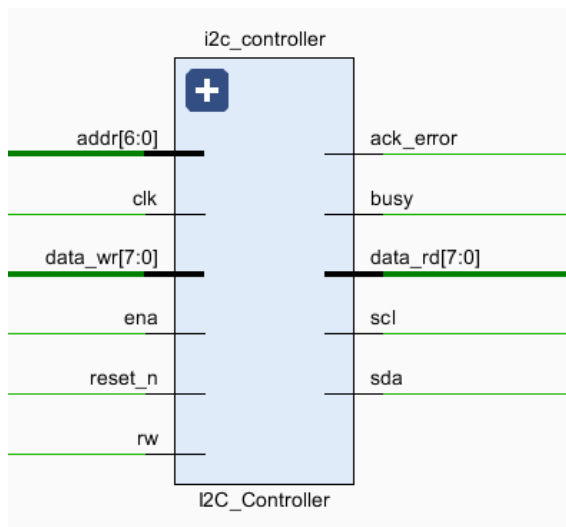


Figura 12 – RTL Schematic I2C Controller

- Clk: clock 100Mhz al placii
- reset\_n: reset general, activ pe 0
- ena: permite sa isi inceapa activitatea
- scl – linia scl bidirectionala
- sda- transmisie date seriale bidirectional
- ack\_error – flag transfer corect/incorect de date
- addr – adresa slave



## UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

- rw – 0 write, 1 read
- data\_wr – date pt scriere pe slave
- data\_rd – date citite de pe slave

Acest modul controlează liniile SDA și SCL pentru magistrala I2C, luând rolul de Master, care permite comunicarea cu periferice slave. Acesta necesită un top level pentru a controla alte operațiuni (configurare sau citire registre din slave). Operațiile efectuate de către master nu se realizează pe ceasul sistemului, ci aceasta își definește un clock intern. Opțiunea de clock stretching este implementată, în cazul în care se folosește un slave care este capabil să țină linia SCL la 0 logic pentru a pune pauză tranzacției de date, deoarece are nevoie de mai mult timp pentru a stoca datele înainte de a continua.

### I2C Slave

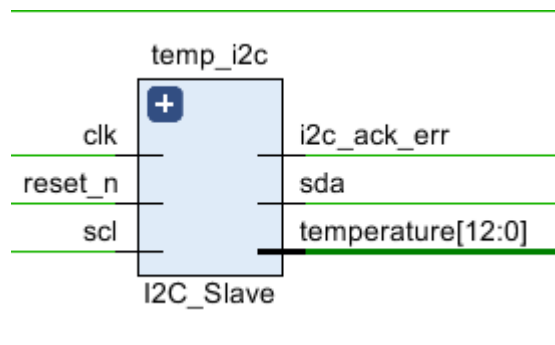


Figura 13 – RTL Schematic I2C Slave

- Clk: clock 100Mhz al placii
- reset\_n: reset general, activ pe 0
- scl – linia scl bidirectionala
- sda- transmisie date seriale bidirectional
- i2c\_ack\_error - flag transfer corect/incorect de date
- temperature(12:0) – temperatura citita pe 13 sau 16 biti

Acest modul permite citirea a 13 sau 16 biti (rezoluția temperaturii fiind generic și configurabilă) a temperaturii.



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

**UART TX**

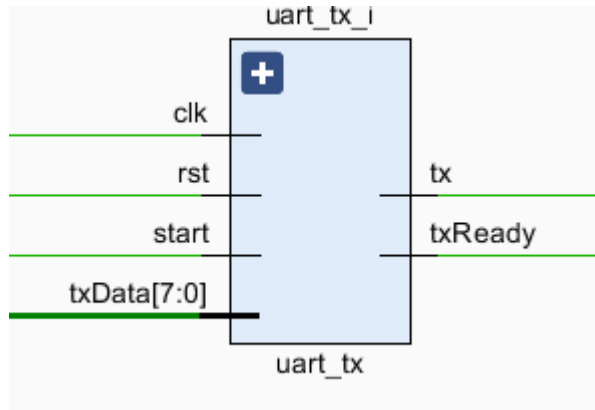


Figura 14 – RTL Schematic uart\_tx

- Clk: clock 100Mhz al placii
- rst: reset
- Start – flag pentru inceperea transmisiei
- txData – byte-ul care urmeaza a fi transmis
- tx – linia pe care sunt transmise datele in mod serial
- txReady – flag ce indica transmitia cu succes a octetului
- 

Acest modul transmite serial, octetul incarcata pe linia txData, precedat de bitul de start, si urmat de bit ul de stop. Transmisia se incepe la activarea semnalului Start, iar daca aceasta s-a finalizat cu succes, se activeaza semnalul txReady.

## UART Send 16

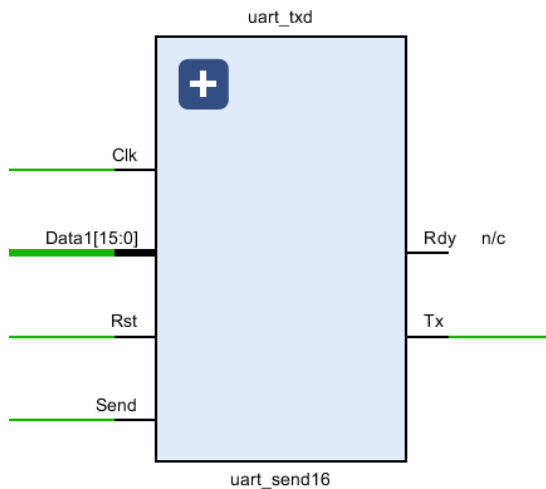


Figura 15– RTL Schematic uart send 16

- Clk: clock 100Mhz al placii
- rst: reset
- Send – flag pentru inceperea transmisiei
- Data1 – octetii care urmeaza a fi transmisi
- tx – linia pe care sunt transmise datele catre modulul uart\_tx
- Rdy – flag ce indica transmitia cu succes a octetilor

Acest modul transmite serial, octetii incarcate pe linia Data1, catre modulul uart tx. Transmisia se incepe la activarea semnalului Send, iar daca aceasta s-a finalizat cu succes, se activeaza semnalul Rdy.





### 4.3 Automatul de stări

#### Automatul de stări pentru I2C Master

Daca se activeaza semnalul de reset, se reseteaza toate semnalele utilizate, si se trece in starea ready. Se asteapta activarea semnalului ena, dupa care se trece in starea command, in cadrul careia se seteaza adresa slave-ului si flagul de rw (0 pentru write, 1 pentru read).

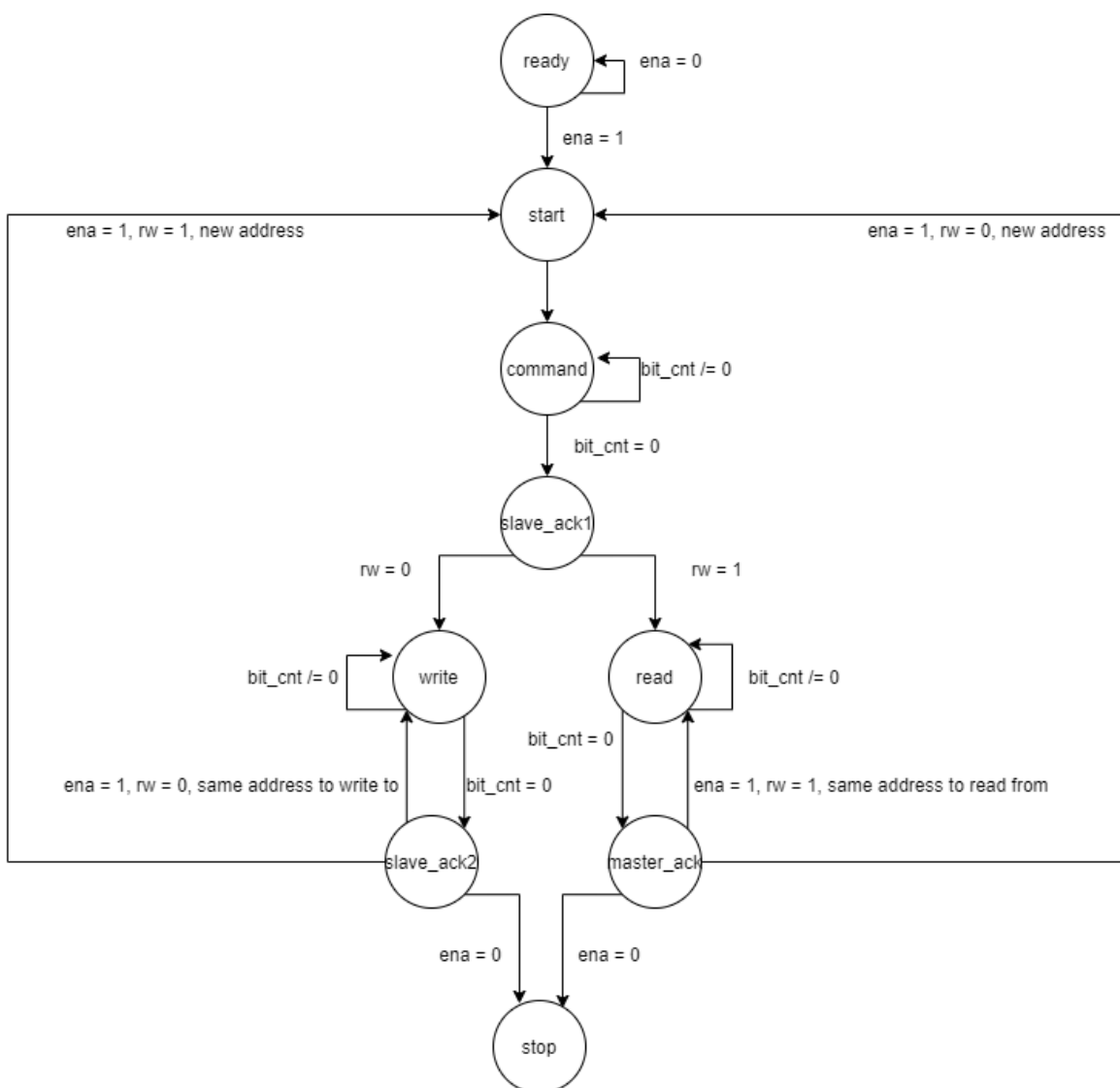


Figura 16 – fsm I2C Controller



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

În starea `slave_ack1`, se așteaptă bitul de acknowledge de la slave, după care, în funcție de flagul `rw`, se trece la scrierea sau citirea din slave. Odată ce una din aceste două stări s-a finalizat, se verifică bitul de acknowledge încă o dată (`slave_ack 2` dacă s-a executat scriere, și `master_ack`, dacă s-a executat citire).

Dacă semnalul `ena` nu este comutat la 0, se continuă cu o altă citire sau scriere, dacă adresa și bitul `rw` rămân neschimbate. Dacă una din cele 2 se modifică, se trece în starea `start`, unde se inițializează cu noile valori.

Dacă semnalul `ena` este comutat la 0, nu se dorește o nouă citire sau scriere, și se trece în starea `stop`, după care în starea `ready`.

### Automatul de stări pentru I2C Slave

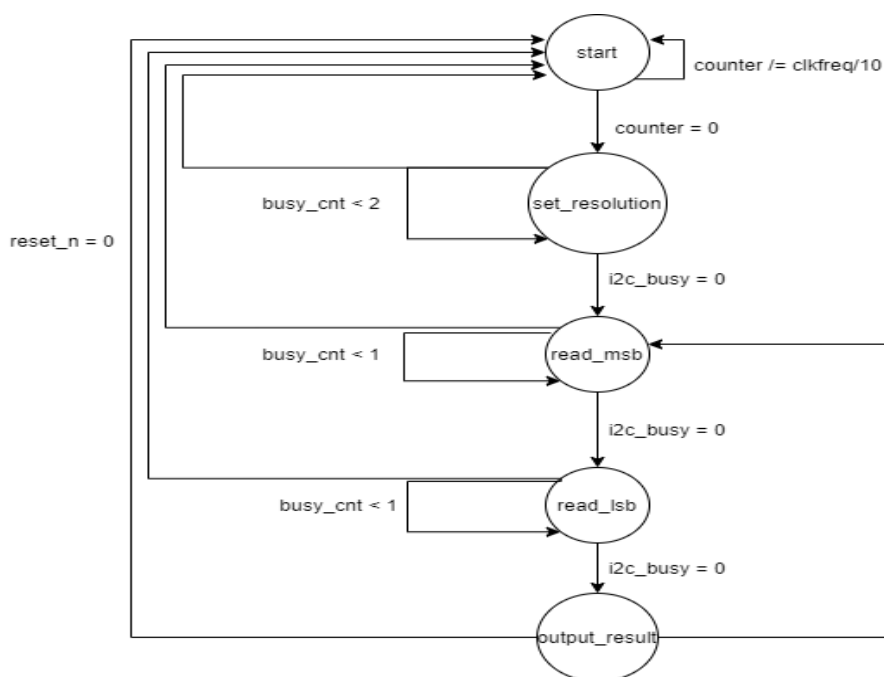


Figura 17 – fsm I2C Slave

Dacă se activează semnalul de reset în oricare din stări, se resetează toate semnalele utilizate, și se trece în starea `start`. În starea `start`, se așteaptă 100ms, pentru a ne asigura că senzorul este gata. Ulterior, se trece în starea `set_resolution`, în care se setează acurătatea temperaturii, pe 13 sau 16 biți, lucru care trebuie setat în registrul de configurare al senzorului, aflat la adresa `0x03`, cât și bitul `rw` setat pe 0 pentru scriere și adresa senzorului. Se trece după în starea `read_msb`, în care se citesc cei mai semnificativi biți ai temperaturii, setând adresa



## UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

registrului 0x00 al senzorului și semnalul rw pe 1. În starea read\_lsb, se citesc cei mai puțin semnificativi biți, setând adresa registrului 0x00 al senzorului și semnalul rw pe 1.

În final, se trece în starea output result, unde se pune pe ieșire temperatura citită, după care se continuă într-o buclă cele 3 stări: read\_msb, read\_lsb, și output result, pentru a actualiza constant temperatura.

Rezoluția temperaturii este generică, și modificarea acesteia duce la modificarea automată a semnalului de ieșire al temperaturii din modul.

### Automatul de stări pentru UART

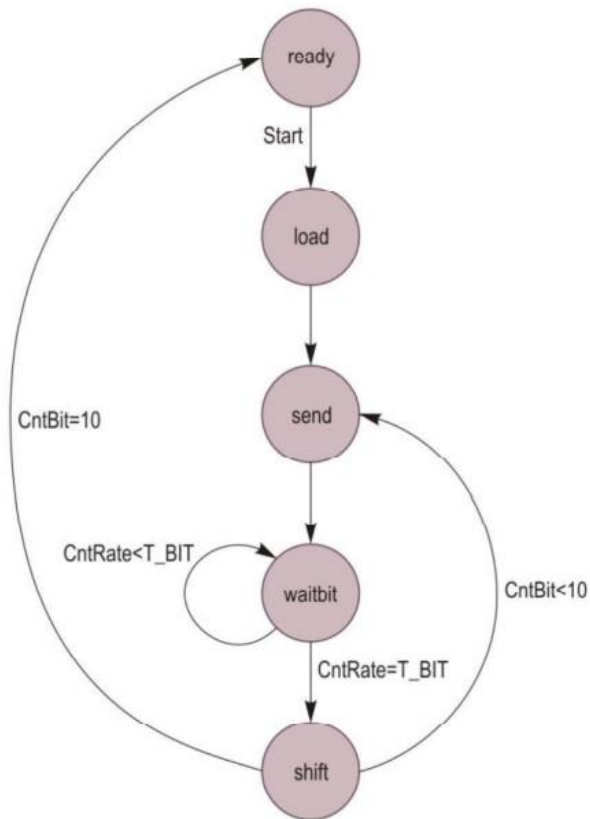


Figura 18 – fsm UART

Acest modul transmite cei 10 biți necesari protocolului UART, bitul de start, octetul dorit, urmat de bitul de stop. În starea waitbit se așteaptă o durată egală cu un bit transmis, după care se shiftează la dreapta registrul de deplasare și se incrementează contorul cntbit. Când s-au transmis toți biții, se revine în starea ready.



#### 4.4 Aplicatia Android

Aplicatia a fost dezvoltata in mediul Android Studio, in limbajul Java. Interfata este una simpla, in care afisam statusul conexiunii bluetooth, sirul de biti receptionat(read buffer), cat si conversia acestuia in zecimal(read buffer int).

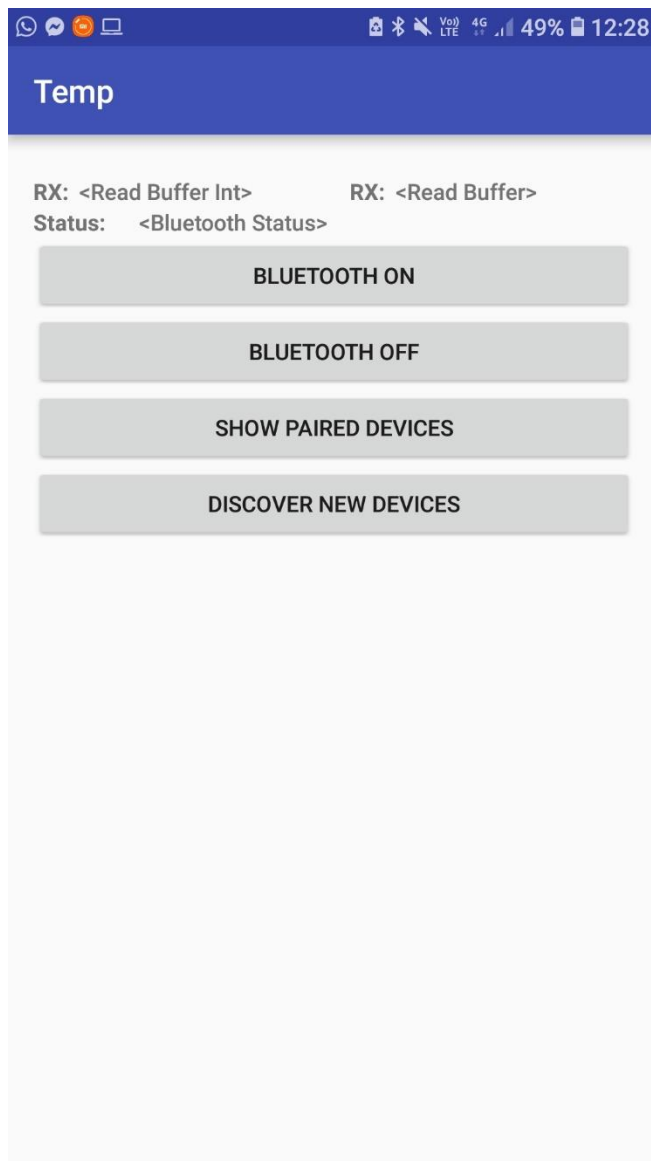


Figura 19 – interfata aplicatie Android

Sub butoanele pentru activarea bluetooth-ului si afisarea dispozitivelor, se vor afisa dispozitivele disponibile, sub forma de lista.



**UNIVERSITATEA TEHNICĂ**  
**DIN CLUJ-NAPOCA**  
**FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE**

Sirul de biti receptionat este separat, ultimii 4 biti cei mai puțin semnificativi, fiind convertiti separat, deoarece reprezinta ceea ce se afla dupa virgula.

## 4.5 Manual de utilizare

Pentru ca orice utilizator sa poată rula proiectul trebuie sa faca urmatoorii pasi:

1. Utilizatorul trebuie sa instaleze Vivado 2020.1
2. Se deschide proiectul cu Vivado
3. Se conecteaza placa FPGA la calculator
4. Se configureaza placa cu fisierul bitstream
5. Se instaleaza aplicatia pe un dispozitiv Android si se conecteaza la modulul bluetooth al placii
6. Se verifica functionalitatea

## 5. Rezultate experimentale

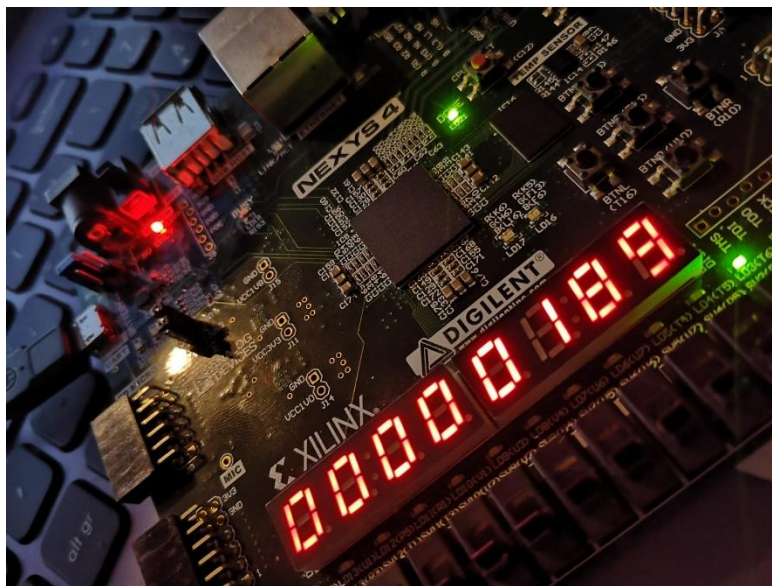


Figura 20 – Test proiect

18.9 in hexazecimal -> 24.9 in zecimal

Pentru testarea proiectului, am folosit placuta Nexys4, pe display-ul careia am afisat temperatura receptionata prin protocolul I2C de la senzorul incorporat.



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

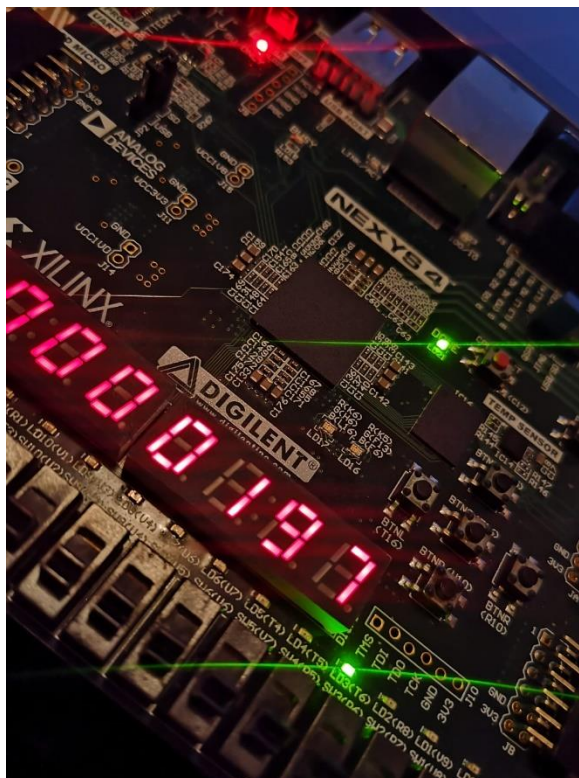


Figura 21 – test proiect după 2 minute de funcționare

19.7 in hexazecimal -> 25.7 in zecimal

Se poate observa că temperatura crește odată cu trecerea timpului, întrucât placa se încălzește. Primele două cifre corespund temperaturii întregi, iar ultima cifră în hexazecimal, corespunde cifrei/cifrelor de după virgulă, așa cum este prezentat și în datasheet-ul senzorului de temperatură



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

## 6. Concluzii

În cadrul acestui proiect s-a realizat implementarea protocolului I2C pentru a comunica cu senzorul de temperatură încorporat în placa FPGA Nexys4DDR.

Înainte de implementare, a fost necesar o înțelegere foarte bună a temei proiectului și studiul amănunțit al interfețelor de comunicare folosite. Pe baza diagramelor de stare s-au implementat cu ușurință modulele care realizează comunicarea și a celor care transmit datele dorite.

S-a prezentat și interfața de comunicare asincronă UART, iar ca dezvoltare ulterioară s-ar putea realiza implementarea protocolului UART și pentru recepția datelor de la modulul bluetooth, spre exemplu pentru a putea selecta rezoluția dorită a temperaturii, pe 13 sau 16 biți.

Pe de altă parte, ca dezvoltare ulterioară a comunicării cu senzorul, s-ar putea utiliza și cele 2 flaguri, Tcrit și Tint, care semnalează depășirea sau scăderea sub un anumit prag de temperatură. Cu ajutorul acestora se poate implementa un termostat care comandă încălzirea sau pornirea răcirii într-o cameră.





**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA  
FACULTATEA DE AUTOMATICA ȘI CALCULATOARE - CATEDRA CALCULATOARE

## 7. Bibliografie

[1] I2C Protocol

<https://www.digikey.com/eewiki/pages/viewpage.action?pageId=10125324>

[2] Temperature sensor datasheet

<http://users.wpi.edu/~riduck/Temp%20Sensor%20Data%20Sheets.pdf>

[4] I2C

<https://www.fpga4fun.com/I2C.html>

[5] Android App Tutorial

<https://www.youtube.com/watch?v=wLRQ9ECIYuA>

[6] Nexys4DDR FPGA Board Reference Manual

[http://users.utcluj.ro/~baruch/resources/Digilent/nexys4ddr\\_rm.pdf](http://users.utcluj.ro/~baruch/resources/Digilent/nexys4ddr_rm.pdf)

[7] UART Protocol

<https://users.utcluj.ro/~baruch/ssc/labor/Testare-Depanare.pdf>