



---

## **TRANSMITEREA UNOR COMENZI DE LA UN DISPOZITIV MOBIL LA O PLACĂ DE DEZVOLTARE**

*Structura sistemelor de calcul*

---

Autori: Munteanu Ionel Andrei și Boca Bianca Malvina

Îndrumător: Lișman Dragoș Florin

Grupa: 30236

FACULTATEA DE AUTOMATICA  
SI CALCULATOARE

13 Decembrie 2022

## Cuprins

<b>1 Rezumat . . . . .</b>	<b>2</b>
<b>2 Introducere . . . . .</b>	<b>3</b>
<b>3 Fundamentare teoretică . . . . .</b>	<b>4</b>
<b>4 Proiectarea și implementarea . . . . .</b>	<b>7</b>
<b>5 Rezultate Experimentale . . . . .</b>	<b>8</b>
<b>6 Concluzii . . . . .</b>	<b>9</b>
<b>7 Bibliografie . . . . .</b>	<b>9</b>

# 1 Rezumat

*Ați fost curioși vreodată cum se pot transmite datele fără fir?* Dacă da, acest proiect vă exemplifică pe scurt bazele comunicării Bluetooth dintre două dispozitive. Proiectul vrea să ilustreze cum putem transmite și recepționa serial datele dintre un telefon mobil Android și o placuță de dezvoltare FPGA (Basys3) prin intermediul unui modul bluetooth Pmod Bt2. Acest proiect a fost realizat cu ajutorul limbajului Vhdl pentru programarea placuței FPGA, dar și a limbajului Java pentru realizarea aplicației de pe mobil și a conexiunii Bluetooth, ambele limbaje de programare îndeplinind atât funcții de back-end cât și funcții de front-end. Prin realizarea cu succes a acestui proiect s-a făcut posibilă transmiterea bidirectională a datelor dintre placuță și aplicație. S-a observat că transmiterea datelor din aplicație pe placuță s-a realizat cu succes în momentul în care datele introduse din aplicație s-au regăsit pe SSD-ul placuței dar și prin aprinderea diodei Led de pe placa de dezvoltare. În schimb, transmiterea datelor de pe placuță în aplicație s-a realizat corect atunci când caracterul aferent codului ASCII format prin intermediul poziționării celor 8 switch-uri pe diferite nivele: HIGH(1 logic) și LOW(0 LOGIC) se va regăsi în aplicație. Această comunicare bidirectională a fost gândită să funcționeze fără conexiune prin cablu atașând placuței FPGA un modul bluetooth care odată conectat la aplicația realizată permite îndeplinirea acestui scop. Ca o concluzie generală asupra acestei teme de proiect putem afirma că realizarea acestuia are doar avantaje întrucât transmiterea informațiilor se realizează în mod eficient și fără resurse suplimentare precum cabluri.

## 2 Introducere

Tema acestui proiect este reprezentată de transmiterea bidirectională a datelor dintre plăcuța Basys3 și aplicația realizată în Android Studio. De fapt, acesta reprezintă scopul principal al acestui proiect împreună cu realizarea conexiunii fără fir prin modulul Pmod BT2. Printre una din tendințele tehnologice la care putem face referire este Metaversul. Metavers este un termen folosit de obicei pentru a descrie conceptul unei iterații viitoare a internetului, alcătuit din spații virtuale 3D partajate, conectate într-un univers virtual perceput Metaversul într-un sens mai larg se poate referi nu numai la lumi virtuale, ci la Internet ca întreg, inclusiv întregul spectru al realității augmentate. Pentru ca o persoană să se "teleporteze" în Metavers are nevoie de cel puțin o pereche de ochelari virtuari și o pereche de căști. Acestea comunică prin intermediul Bluetooth ului. Tocmai de aceea, conexiunile prin Bluetooth reprezintă un început major pentru realizarea unui viitor tehnologizat mult mai eficient.

Domeniul de studiu se focusează în principiu pe realizarea unei conexiuni între hardware și software îmbinând două limbaje de programare "cu găndire" total diferită: vhdl și java.

Printre termenii de specialitate folosiți pentru a înțelege modul de funcționare a proiectului putem face referire la:

- UART - Universal Asynchronous Receiver-Transmitter. Se ocupă cu transmisarea datelor serial, adică bit cu bit și independent față de un ceas.
- RX - modul pentru recepția datelor
- TX - modul pentru transmiterea datelor
- state Machine - implementează logica fiecărui modul descris în proiect
- BluetoothSocket - cel care se ocupă cu stabilirea conexiunii între aplicație și modul
- BluetoothInputSocket - reprezintă locul din care se vor scrie datele pe plăcuță
- BluetoothOutputSocket - reprezintă locul în care se vor scrie datele de pe plăcuță
- Handler - se ocupă cu sincronizarea threadului principal al aplicației cu celelalte threaduri.
- TextView, Button, EditText, ButtonImage, Image - elemente definitorii ale aplicației.

Printre obiectivele principale ale acestui proiect putem menționa

- realizarea unei aplicații care să comunice cu plăcuță
- realizarea unei conexiuni Bluetooth între plăcuță și aplicație
- transmiterea datelor de pe plăcuță în aplicație
- recepționarea datelor din aplicație și afișarea lor pe SSD.
- realizarea unei conexiuni necablate și mult mai eficiente

În următoarele paragrafe se vor explica mult mai pe larg cum funcționează acest proiect. Astfel:

- **Fundamentare teoretică** - se prezintă noțiunile de bază și principiile pe care acest proiect se bazează.
- **Proiectarea și implementarea** - partea în care se prezintă etapele parcuse pentru atingerea obiectivelor propuse. În această secțiune regăsim: Schema bloc, soluția proiectului, detaliile de implementare, metoda experimentală utilizată.
- **Rezultatele experimentale** - Această etapă cuprinde rezultatele simulării surprinse prin intermediul fotografiilor. De asemenea, tot în cadrul acestui subiect sunt prezentate și dificultățile întâmpinate pe parcursul realizării proiectului.
- **Concluzii** - Sumar al raportului proiectului.

### 3 Fundamentare teoretică

Pentru realizarea acestui proiect a fost nevoie de dobândirea unor cunoștiințe atât în vederea implementării părții de transmisie serială în vhdl cât și a aplicației. Pentru realizarea aplicației ne-am folosit de Android Studio, un program bazat pe Java care servește realizării aplicațiilor.

#### 1. REALIZAREA APLICAȚIEI

Aplicația, realizată în Android Studio, în esență nu a fost foarte dificil de realizat deoarece, o dată cu înțelegerea programului utilizat, se poate crea cu ușurință aceasta. Pentru realizarea celor două layout-uri ne-am folosit de elementele definitorii existente deja în aplicație. În acestea regăsim Button, TextView, EditText, ImageButton.

- Button - element definitoriu care permite utilizatorului să lanseze o acțiune, iar pentru realizarea acestui lucru se suprascrie metoda *button.setOnClickListener()* oferind posibilitatea ca la apăsare să se realizeze acțiunea dorită.
- TextView - permite inserarea textului fără a acorda permisiunea ca textul să fie modificat de către utilizator. Pentru a seta textul folosim metoda *textView.setText(String dorit)*
- EditText - permite atât inserarea unui text cât și modificarea acestuia de către utilizator.
- ImageButton - în sine este un button care performă acțiuni singura diferență fiind că acest Button are atașată o imagine și nu denumirea lui.

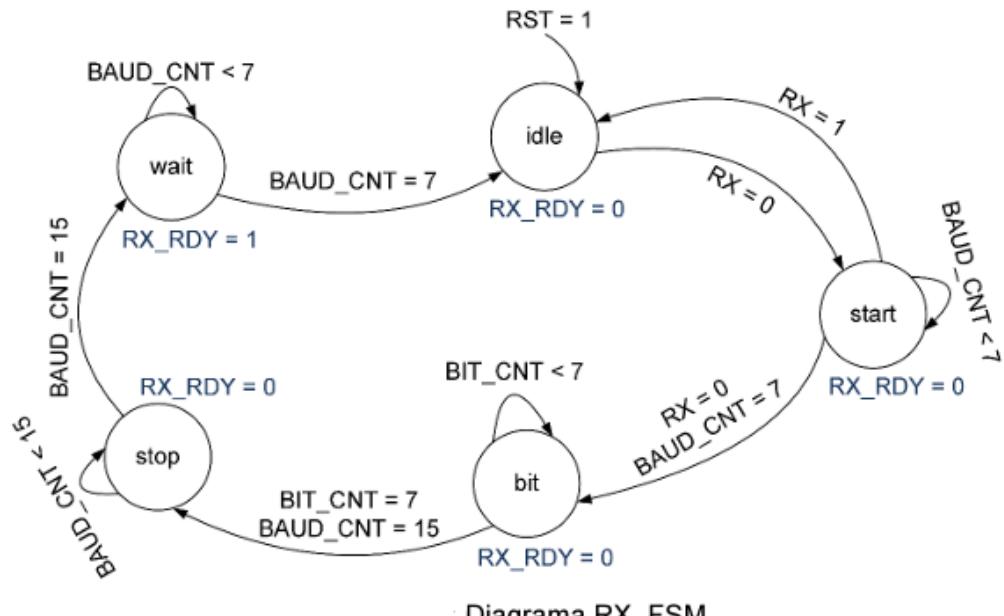
Pe lângă aceste aspecte vizibile ale aplicației a fost nevoie de cunoașterea noțiunilor de bază ale clasei Bluetooth pentru realizarea conexiunii. În vederea realizării acestui obiectiv avem nevoie de noțiunile teoretice ale elementelor necesare din clasa Bluetooth precum: BluetoothSocket, BluetoothInputStream, BluetoothOutputStream.

BluetoothSocket - se ocupă de realizarea conexiunii dintre două dispozitive. Se apelează o funcție foarte esențială *BluetoothDevice.createRfcommSocketToServiceRecord()* care va returna o metodă care trebuie invocată asupra device-ului. O dată ce conexiunea a fost stabilită vor fi disponibile InputStream dar și OutputStream, folosite pentru realizarea obiectivelor dorite de scriere și de citire.

#### 2. REALIZAREA TRANSMITERII SERIALE

Pentru a putea realiza transmiterea datelor și recepționarea acestora este nevoie de cunoașterea modului de funcționare atât a plăcuței Basys3 cât și a modulului Bluetooth. Modulul Bluetooth atașat plăcuței are 12 pini dintre care se vor folosi doar pinii aferenți lui RX și TX prin intermediul cărora se face posibilă această comunicare bidirectională dintre plăcuță și dispozitiv. Comunicarea UART (Universal Asynchronous Receiver Transmitter) este metoda folosită pentru realizarea scopului dorit. Această metodă are la bază două module RX și TX, primul reprezentând receptia iar al doilea reprezentând transmisia. Cu ajutorul acestora, biții de date sunt transferați succesiv de-a lungul unui canal de comunicare sau magistrală. La fiecare comunicare se transmite un octet (8 biți), bitul de start și bitul de stop. Bitul de start este folosit pentru a putea semnala începerea transmiterii celor 8 biți de date iar bitul de stop semnalează terminarea transmiterii acestora. Datorită faptului că comunicarea este asincronă (comunicarea se realizează independent de un ceas comun), se folosește o rată de eşantionare, generată independent la sursă și destinație, numită baud rate (număr de biți transmiși pe secundă). Valorile uzuale pentru baud rate sunt 9600. Practic, un bit este valid pe linia de transmisie pentru un interval dat de timp, egal cu inversul baud rate. Pentru fiecare modulul RX și TX folosim câte un baud rate separat. Numărul de caractere transmise se obțin împărțind baud-rate-ul la numărul de biți transmiși (în cazul nostru avem nevoie de 10 biți pentru ca un caracter să fie transmis).

a) Diagrama de stare pentru RX:



Conform îndrumătorului de laborator al Arhitecturii Calculatoarelor, diagrama de stare pentru RX, poate fi interpretată în felul următor:

Cei 8 biți citiți se vor stoca în RX DATA care este un semnal pe 8 biți. RX RDY este setat pe 1 după ce s-a terminat recepția unui caracter. În RX FSM sunt necesare două semnale auxiliare cu funcționalitate de numărător: BITCNT și BAUDCNT. Semnalul BITCNT are o funcționalitate la fel ca în TXFSM, de numărător în interiorul FSM-ului, el reprezentând poziția din RXDATA unde se stochează bitul curent. El trebuie incrementat în starea bit, de fiecare dată când se ajunge la mijlocul bitului curent, și trebuie resetat după fiecare caracter transmis (se poate reseta în starea idle sau în toate stările exceptând bit). Semnalul BAUDCNT contorizează câte activări (valori de 1) are semnalul BAUD ENable, el fiind utilizat pentru a verifica dacă s-a ajuns în mijlocul intervalului de bit sau la sfârșit de interval. În mare, funcționarea RXFSM este următoarea:

- În starea idle, BAUDCNT este setat pe 0, se trece în start dacă pe linia serială apare 0 (RX=0).
- În starea start, BAUDCNT este incrementat și folosit pentru a eșantiona linia serială (RX). Dacă s-a ajuns la mijlocul bitului de start (BAUDCNT=7 și RX=0) atunci BAUDCNT este resetat și se trece la starea bit.
- În starea bit, BAUDCNT este implicit incrementat, până ajunge la valoarea 15 (mijlocul bitului curent). Dacă a ajuns la 15, se salvează bitul curent (RXDATA[BITCNT]=RX), se resetează BAUDCNT (se reîncepe numărarea până la 15) și se incrementează BITCNT. Dacă BITCNT=7 și BAUDCNT=15, atunci sunt resetate și se trece în starea stop.
- În stop se așteaptă să treacă încă un interval de bit pe linia serială, interval care acoperă practic jumătate din ultimul bit de date și jumătate din bitul de STOP (amintiți-vă că s-a început de la jumătatea bitului de START, deci în stop s-a sărit din bit la jumătatea ultimului bit de date).
- În wait se semnalează că s-a recepționat un caracter (RXRDY=1), se mai așteaptă o jumătate de interval de bit (să se termine bitul de STOP) și se trece în idle.

b) Diagrama de stare pentru TX:

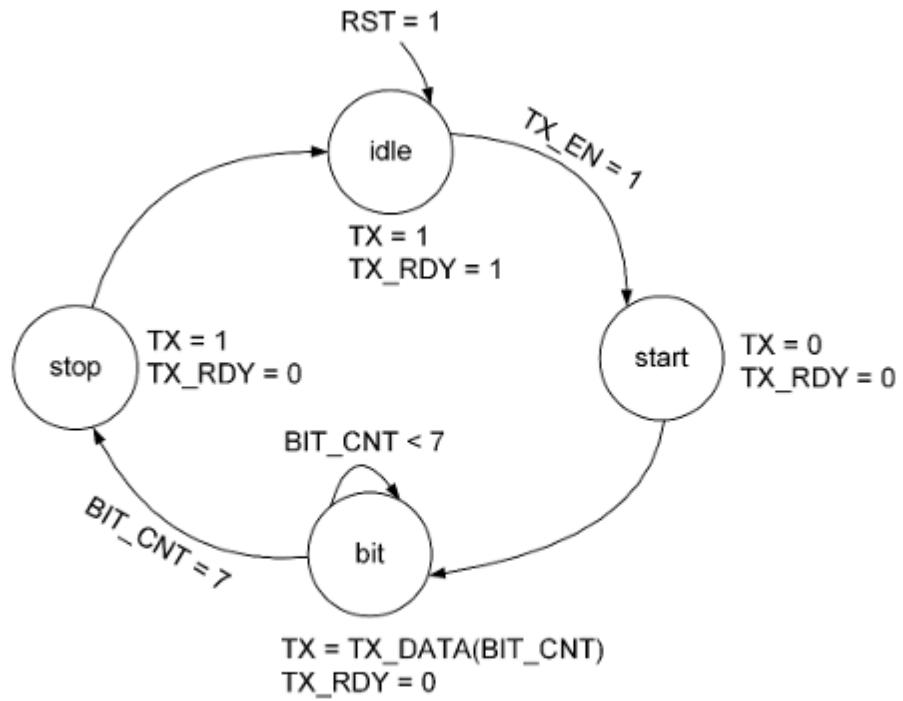


Diagrama TX\_FSM

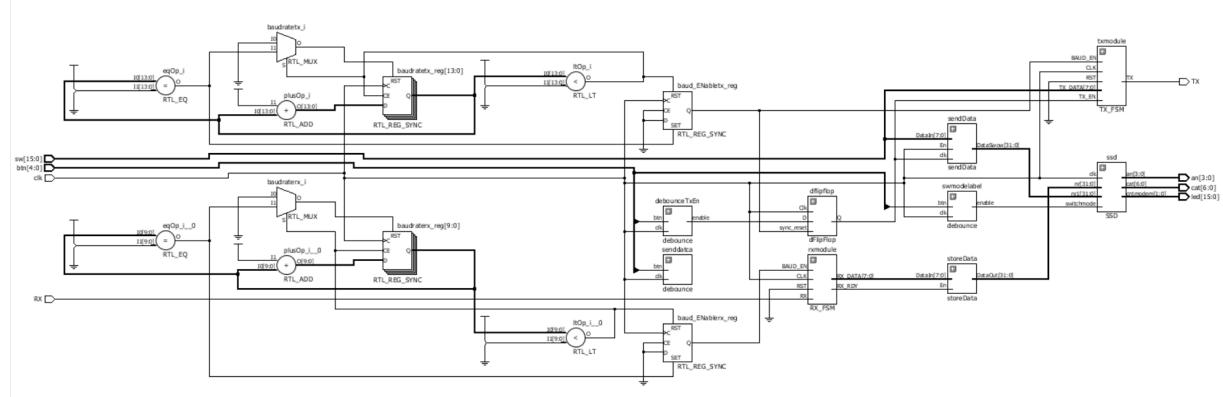
Conform îndrumătorului de laborator al Arhitecturii Calculatoarelor, diagrama de stare pentru TX, poate fi interpretată în felul următor:

Diagrama în detaliu a TXFSM este prezentată implementând protocolul de transmisie serială pentru caractere de 8 biți. O tranziție între stări poate avea loc pe frontul crescător de ceas doar dacă BAUDENable este ‘1’. Astfel, se asigură că un bit rămâne pe linia de transmisie pentru intervalul dat de baud rate. Semnalul BITCNT are o funcționalitate de numărător în interiorul FSM-ului, el reprezentând poziția bitului curent de transmis din TXDATA. El trebuie incrementat în starea bit și trebuie resetat după fiecare caracter transmis (se poate reseta în starea idle sau în toate stările exceptând bit).

## 4 Proiectarea și implementarea

Diagrama de mai jos reprezintă schema generală a proiectului. Principalele module regăsite de mai sus sunt:

1. TXFSM - se ocupă de transmiterea datelor. Acesta se conectează la TX-ul celuilalt dispozitiv. ;
2. RX - se ocupă de receptia datelor. Acesta se conectează la RX-ul celuil dispozitiv;
3. SSD - necesar pentru a putea afișa caracterele dorite pe plăcuță. Acesta este conceput în felul următor. La apăsarea unui buton, cel de sus (btnU) se comută pe display între cele două modalități de comunicare, pentru a ajuta utilizatorul să știe ce date au fost transmise sau recepționate. La fiecare apăsare a butonului BTNC, în modul de transmitere, se trimit caracterul către dispozitiv.

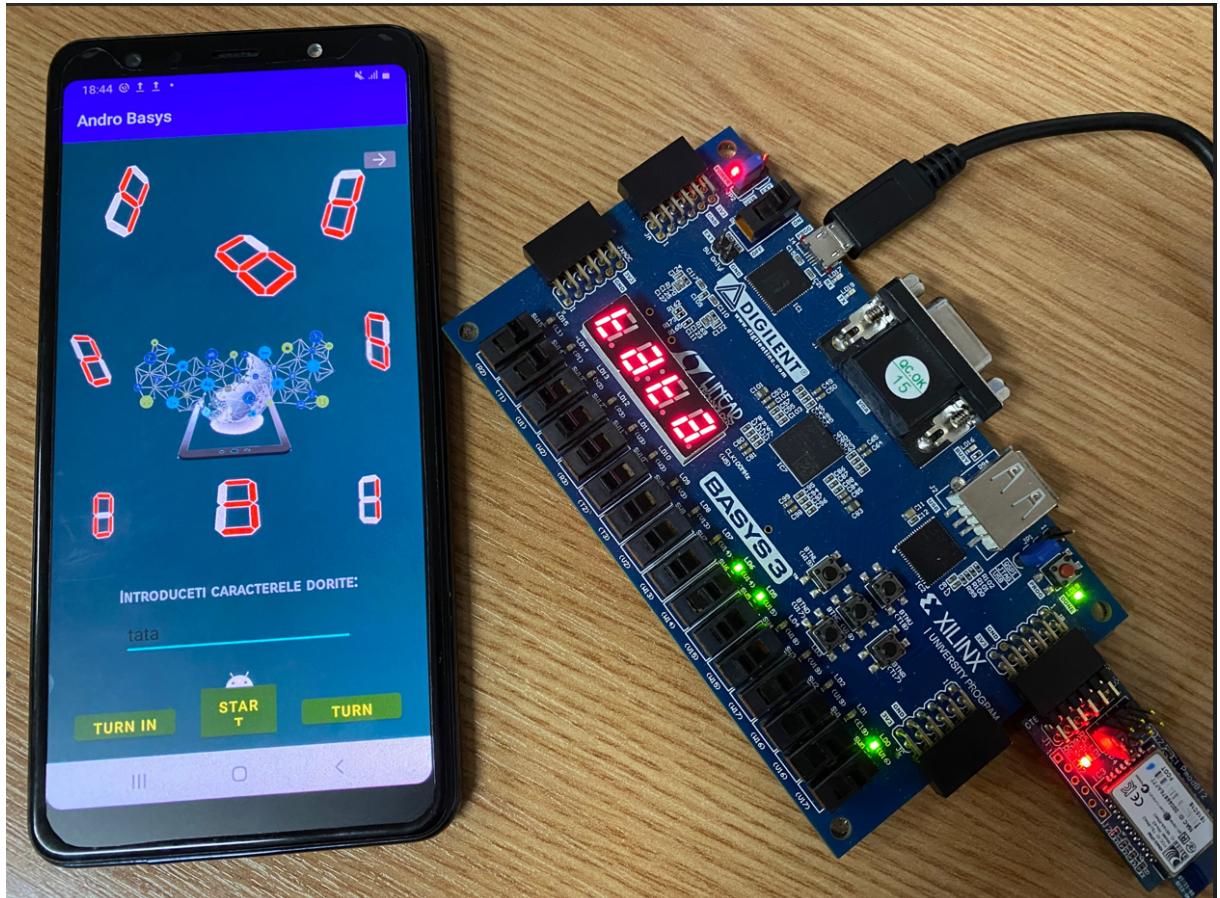


## 5 Rezultate Experimentale

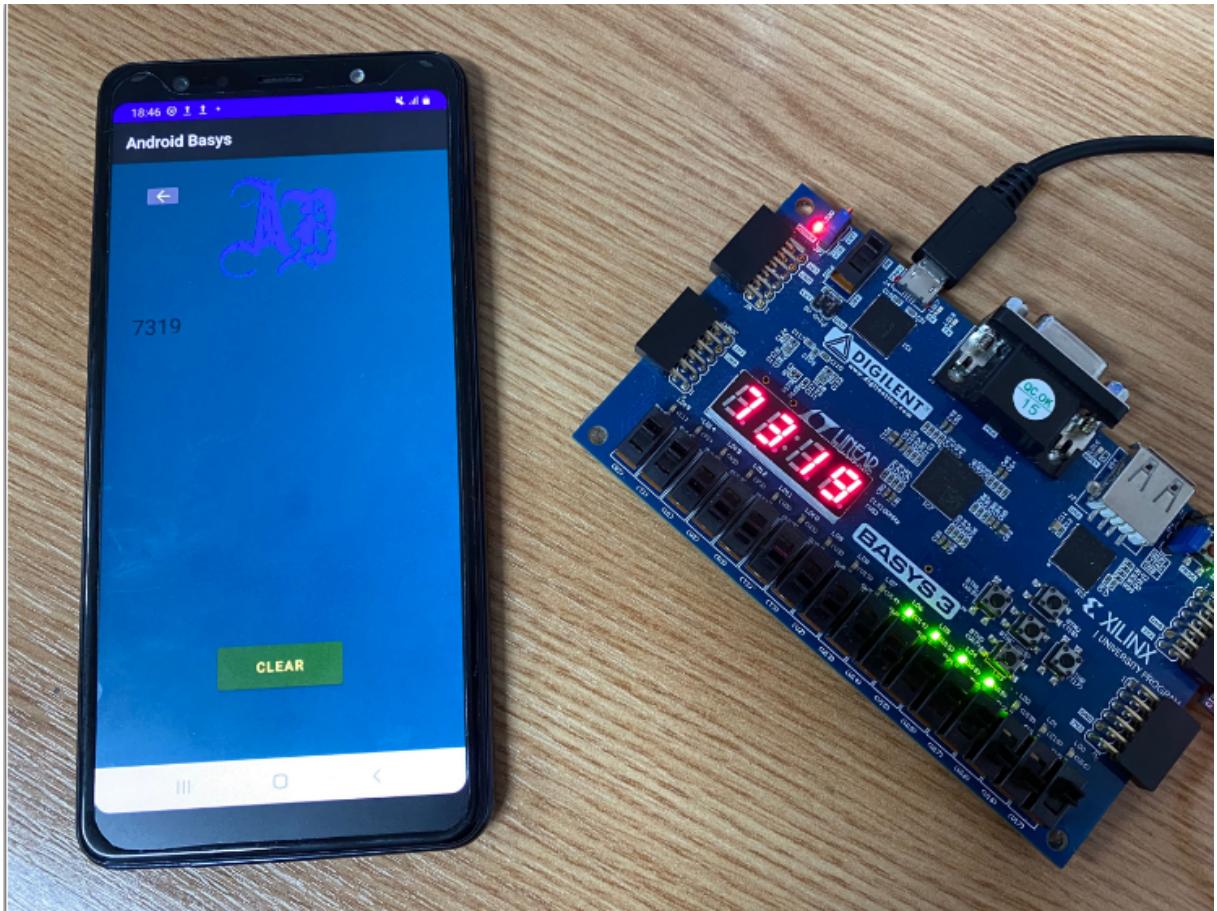
Pentru a demonstra funcționalitatea acestui proiect vom atașa o serie de poze atât pentru transmisie cât și pentru recepție.

Pe parcursul acestui proiect am întâlnit o dificultate care se poate rezuma la conexiunea Bluetooth întrucât, atunci când treceam din MainActivity în SecondActivity în aplicație se pierdea conexiunea Bluetooth iar transmiterea datelor de pe placă în aplicație nu a mai fost posibilă în acest fel.

Transmisia datelor de pe placă în aplicație:



Transmiterea datelor din aplicație pe dispozitiv:



## 6 Concluzii

Ca o concluzie generală asupra acestui proiect putem afirma că dificultatea proiectului a fost una medie întrucât am avut nevoie de dobândirea multor cunoștințe în plus pentru realizarea transmiterii și a aplicației. Nelucrând deloc în Android Studio, a fost nevoie de cunoașterea acestui program pentru a da startul acestui proiect. Deși timpul dedicat realizării acestui proiect a fost destul de mare, rezultatele obținute sunt unele pe măsură. Satisfacția oferită de rezultatele proiectului este majoră deoarece putem vedea cu ochiul liber și aproape imediat cum datele "zboără" de la un dispozitiv la altul. Când are loc deschiderea aplicației aceasta se va conecta automat la Bluetooth ul atașat plăcuței FPGA. Din acest moment se poate face posibilă transmiterea datelor de la plăcuță la aplicație și invers. Când datele din aplicație sunt transmise spre plăcuță vor putea fi observate imediat pe SSD conform codificărilor caracterelor respective bazate pe codul ASCII, iar când datele de la plăcuță sunt transmise de la plăcuță spre aplicație, fiecare caracter transmis la apăsarea de buton poate fi văzut în aplicație.

## 7 Bibliografie

[https://users.utcluj.ro/~onigaf/files/teaching/AC/AC\\_indrumator\\_laborator.pdf](https://users.utcluj.ro/~onigaf/files/teaching/AC/AC_indrumator_laborator.pdf)

<https://users.utcluj.ro/~baruch/ro/pages/cursuri/structura-sistemelor-de-calcul/laborator.php>

<https://developer.android.com/>