



УНИВЕРЗИТЕТ
У НОВОМ САДУ



ФАКУЛТЕТ
ТЕХНИЧКИХ НАУКА

Трг Доситеја Обрадовића 6, 21000 Нови Сад, Југославија
Деканат: 021 350-413; 021 450-810; Централa: 021 350-122
Рачуноводство: 021 58-220; Студентска служба: 021 350-763
Телефакс: 021 58-133; e-mail: ftndeans@uns.ns.ac.yu



Сертификован
систем
квалитета



RAČUNARSKA ELEKTRONIKA

NAZIV PROJEKTA:

Tic Tac Toe

TEKST ZADATKA:

Implementacija igrice "Iks-Oks" i kreiranje pratećeg GUI-a (Graphic User Interface) u Qt razvojnom okruženju na RaspberryPi pločici uz pomoć matrične tastature

Mentor:

Prof. Ivan Mezei

Student:

Ivan Milin EE179/2019

U Novom Sadu, 11.07.2023.

Sadržaj

1	Uvod	2
2	Matrična tastatura	3
3	Korisnički interfejs	4
4	Funkcije u kodu	5
4.1	<code>void init_keypad();</code>	5
4.2	<code>char get_key();</code>	5
4.3	<code>char whichTurn(int *counter);</code>	5
4.4	<code>void button_pressed();</code>	5
4.5	<code>void luckyWinner();</code>	5
4.6	<code>void clearMatrix();</code>	6
4.7	<code>void on_pushButton_clicked();</code>	6
4.8	<code>void on_pushButton_17_clicked();</code>	6
5	Zaključak	7

1 Uvod

RaspberryPi prikazan na slici [1](#) je *embedded* računar prvenstveno namenjen u edukativne svrhe i za entuzijaste koji žele da se uđu u svet programiranja.

Ono što ga čini interesentnim su cena, male dimenzije (8,6cm x 5,4cm x 1,7cm) i mogućnost priključivanja ne standardnih dodatnih komponenti i opreme.

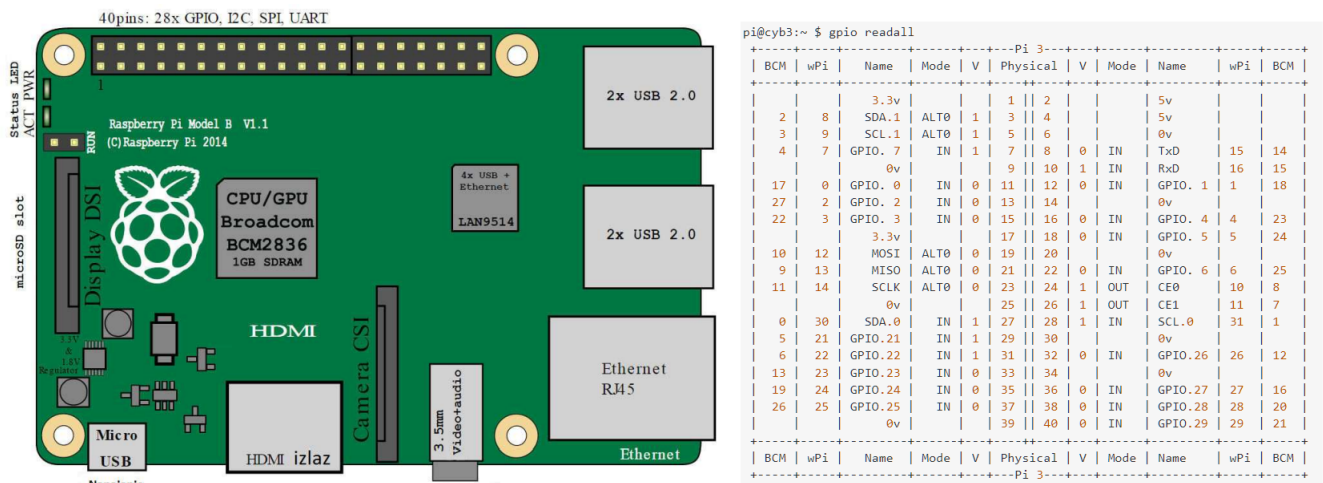
Tema projekta je implementacija kompjuterske igrice "Iks-Oks" i kreiranje pratećeg GUI-a (*Graphic User Interface*) u Qt razvojnom okruženju na RaspberryPi pločici.

Kao periferija korišćena je matrična tastatura koja je priključena na RaspberryPi pomoću 40-pinskog GPIO(*eng. General Purpose Input/Output*).

Kako bi moglo da se radi sa GPIO pinovima neophodno je instalirati odgovarajuće biblioteke, u ovom projektu korišćena je "WiringPi"biblioteka.

Za potrebe povezivanja matrične tastature sa RaspberryPi iskorišćeni su fizički pinovi 11,13,15,19 kao i pinovi 12,16,18,22, listing pinova na slici [1](#).

Na slici [2](#) prikazano je kako su RaspberryPi i matrična tastatura povezani.



Slika 1: RaspberryPi 4 i listing GPIO pinova



Slika 2: Povezivanje RaspberryPi i matrične tastature

2 Matrična tastatura

Zbog ograničenog broja pinova, najčešće nije moguće povezati sve tastere direktno sa RaspberryPi.

Jedna od tehnika povezivanja pomoću koje možemo uštedeti broj iskorišćenih pinova je *matrično povezivanje*, u slučaju da su tasteri raspoređeni u matricu $n \times m$.

Na taj način zauzima se samo $n+m$ pinova, a moguće je očitati stanja $n \cdot m$ tastera, znači 8 pinova biće iskorišćeno za očitavanje 16 tastera.

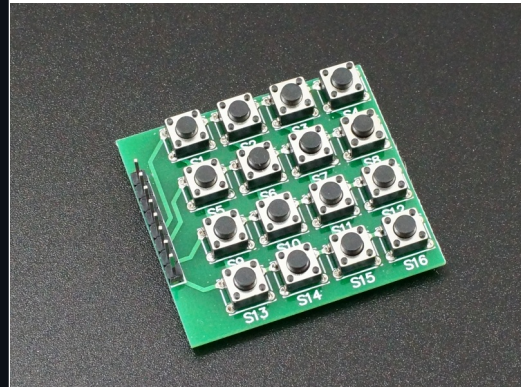
Sam izgled matrične tastature prikazan je na slikama 2. i 3b.

Na slici 3a. prikazan je kod pomoću kog je prvo testirano da li je dobro povezan RaspberryPi sa matričnom tastaturom.

Na početku ni jedan taster nije pritisnut, sve vrste postavljene su u stanje visoke impedanse, dok su kolone u stanju logičke jedinice preko pull-up otpornika.

Zatim se proverava koji taster je pritisnut i po aktiviranju spaja se data kolona sa vrstom i ispisuje ASCII karakter u terminal.

```
17 void setup() {
18     wiringPiSetup();
19
20     for (int i = 0; i < ROWS; i++) {
21         pinMode(rowPins[i], OUTPUT);
22         digitalWrite(rowPins[i], HIGH); // Set row pins to default high
23     }
24     for (int j = 0; j < COLS; j++) {
25         pinMode(colPins[j], INPUT);
26         pullUpDnControl(colPins[j], PUD_UP); // Enable pull-up resistors on column pins
27     }
28 }
29 char getKey() {
30     for (int i = 0; i < ROWS; i++) {
31         digitalWrite(rowPins[i], LOW); // Set current row low
32
33         for (int j = 0; j < COLS; j++) {
34             if (digitalRead(colPins[j]) == LOW) {
35                 // Button in row i and column j is pressed
36                 while (digitalRead(colPins[j]) == LOW); // Wait for key release
37                 digitalWrite(rowPins[i], HIGH); // Restore row to high
38
39                 return keys[i][j]; // Return the pressed key
40             }
41         }
42         digitalWrite(rowPins[i], HIGH); // Restore row to high
43     }
44 }
45
46 return "\0"; // No key pressed
47 }
```



(a) .

(b) .

Slika 3: Test program i matrična tastatura

3 Korisnički interfejs

GUI (Graphic User Interface) predstavlja vizuelnu vezu pomoću koje se ostvaruje lakša interakcija između korisnika i sistema.

Na slici 4 prikazana je realizacija GUI-a koji se koristi u projektu.

Najveću deo čini tabela koja predstavlja polja u koju će biti upisivani karakteri "X" ili "O" koji se generišu pri pritisku tastera matrice tastature

Ukoliko se dogodi da se četiri ista karaktera zadese u dijagonali ili unutar iste kolone ili reda jedan od igrača postaje pobednik, ta polja postaju crvena 5.

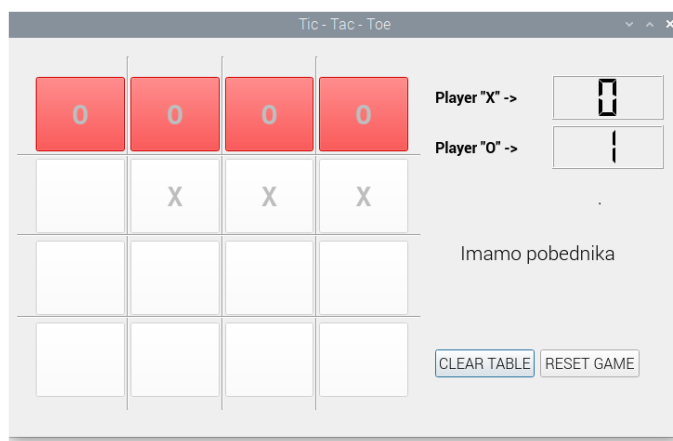
Takođe brojač koji se nalazi u desnoj gornjoj strani GUI-a biće uvećan za jedan 5.

Taster *CLEAR TABLE* tipa *PushButtons* predviđen je da se pritisne kada jedan od igrača pobedi kako bi se polja vratila u inicijalnu vrednost, dok je taster *RESET GAME* tipa *PushButtons* predviđen je da se pritisne kad oba igrača odluče da je kraj igri.

Pri svakom pritisku tastera menja se tekst u labeli : "Na redu je igrač O" ili "Na redu je igrač X".



Slika 4: Korisnički interfejs



Slika 5: Ispis pobednika

4 Funkcije u kodu

U nastavku je dat opis svake od funkcije koja je realizovana unutar projekta.

4.1 *void init_keypad();*

Opis: Na početku ni jedan taster nije pritisnut, sve vrste postavljene su u stanje visoke impedanse, dok su kolone u stanju logicke jedinice preko pull-up otpornika.

Parametri: Nema

Povratna vrednost: Nema

4.2 *char get_key();*

Opis: Proverava koji je taster pritisnut.

Parametri: Nema

Povratna vrednost: ASCII kod koji predstavlja taster, ili vrednost 0 ukoliko nije pritisnut ni jedan taster

4.3 *char whichTurn(int *counter);*

Opis: parametar *counter* uvećava se svaki put kad se pritisne taster i pomoću tog parametara određujemo koji igrač je na redu da igra.

Parametri: Broj koji je celobrojan

Povratna vrednost: ASCII kod koji predstavlja koji igrač je na redu da igra.

4.4 *void button_pressed();*

Opis: U zavisnosti od tastera koji je pritisnut prosleđuje se vrednost u određeno polje GUI-a.

Parametri: Nema

Povratna vrednost: Nema

4.5 *void luckyWinner();*

Opis: Proverava koja polja su pritisnuta, ukoliko se dogodi da se četiri ista karaktera zadese u dijagonali ili unutar iste kolone ili reda ta polja postaće crvena, brojač koji se nalazi u desnoj gornjoj strani GUI-a biće uvećan za jedan.

Parametri: Nema

Povratna vrednost: Nema

4.6 *void clearMatrix();*

Opis: Briše sve vrednosti iz tabele.

Parametri: Nema

Povratna vrednost: Nema

4.7 *void on_pushButton_clicked();*

Opis: Poziva funkciju *clearMatrix()*, postavlja promenljivu *counter* na nula i ispisuje poruku u labelu koja se nalazi u GUI-u.

Parametri: Nema

Povratna vrednost: Nema

4.8 *void on_pushButton_17_clicked();*

Opis: Ova funkcija se poziva kada želimo da krenemo ispočetna.

Poziva funkciju *clearMatrix()*, postavlja sve bitne promenljive na nula i ispisuje poruku u labelu koja se nalazi u GUI-u.

Parametri: Nema

Povratna vrednost: Nema

5 Zaključak

Prilikom testiranja uklonjene su sve greške koje su se pojavile.

U prilogu ispod nalazi se link ka github repozitorijumu na kome se nalazi projekat iz predmeta "Računarska elektronika"

https://github.com/ivanMilin/RPI_projekat.git