

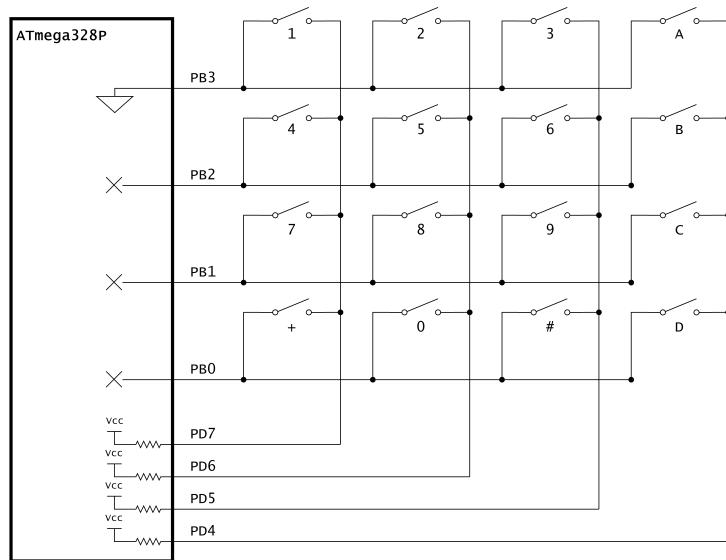
Poglavlje 19

Matrična tastatura i analogni džojstik

19.1 Matrična tastatura

Kod tastatura koje sadrže veći broj tastera najčešće nije praktično izvršiti direktno povezivanje tastera na pinove mikrokontrolera, zbog ograničenog broja pinova. Jedna od najčešće korišćenih tehnika povezivanja u ovom slučaju jeste tehnika *matričnog povezivanja*, gde su tasteri raspoređeni u matricu dimenzija $m \times n$. Na ovaj način je, pomoću $m + n$ pinova mikrokontrolera, moguće očitati stanje ukupno $m \cdot n$ tastera. Na slici 19.1, prikazana je shema povezivanja matrične tastature dimenzija 4×4 na portove B i D mikrokontrolera ATmega328P. Sam izgled matrične tastature je dat na slici 19.2. Tasteri na ovakovom tipu tastatura su uglavnom membranski, međusobno povezani provodnim vezama u skladu sa već prikazanom internom shemom.

Skeniranje, odnosno očitavanje stanja tastera vrši se sekvencialno, vrstu po vrstu. U cilju jednostavnijeg objašnjenja, usvojiće se sledeći način povezivanja matrične tastature. Niža četiri bita porta B (pinovi PB3...PB0) predstavljaju vrste, dok se viša četiri bita porta D (pinovi PD7...PD4) koriste kao kolone. Vrsta se aktivira postavljanjem logičke 0 na odgovarajući pin porta B, dok su ostali pinovi iz ove grupe u stanju visoke impedanse. Viša 4 bita porta D koriste se kao ulazi, preko kojih se očitavaju stanja tastera u okviru trenutno aktivne vrste. Taster koji je trenutno pritisnut kratko spaja odgovarajući ulaz na logičku nulu, dok, u suprotnom, interni *pull-up* otpornik postavlja stanje linije na jedinicu. Logika skeniranja tastature prikazana je u tabeli 19.1.



Slika 19.1: Interna shema povezivanja matrične tastature sa 16 tastera

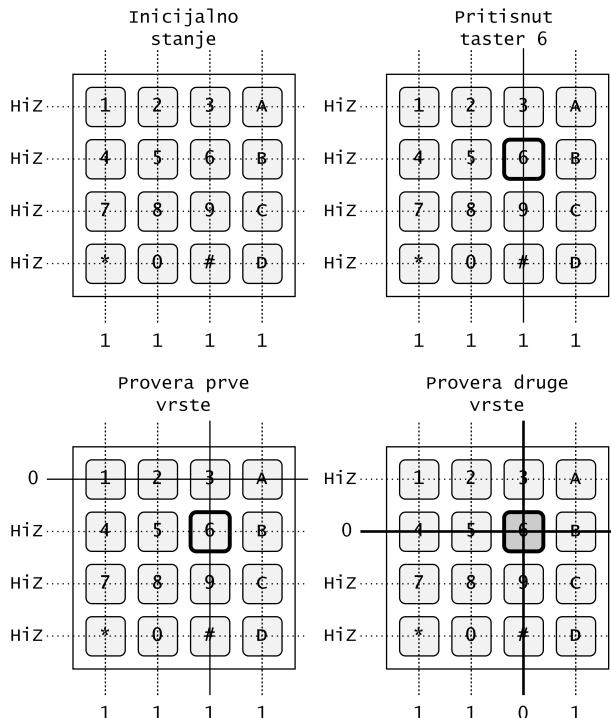


Slika 19.2: Izgled matrične tastature sa 16 tastera

Tabela 19.1: Logika skeniranja tastature

Vremenski interval	Izlazi za aktiviranje vrste				Ulazi za očitavanje stanja tastera			
	PB3	PB2	PB1	PB0	PD7	PD6	PD5	PD4
T1	0	HiZ	HiZ	HiZ	1	2	3	A
T2	HiZ	0	HiZ	HiZ	4	5	6	B
T3	HiZ	HiZ	0	HiZ	7	8	9	C
T4	HiZ	HiZ	HiZ	0	*	0	#	D

Na slici 19.3 je ilustrovan način skeniranja matrične tastature za slučaj kada je pritisnut taster 6. Naime, u inicijalnom stanju, kada nijedan taster nije pritisnut, sve vrste su postavljene u stanje visoke impedanse (HiZ), dok su, nasuprot tome, kolone postavljane na visok logički nivo preko *pull-up* otpornika. Kada je taster pritisnut (u ovom slučaju taster 6), data kolona se aktivira i kratko spaja sa datom vrstom. Sa druge strane, vrste se ciklično skeniraju postavljanjem na logičku nulu tokom odgovarajućeg vremenskog intervala, kao što je moguće i videti u donjem redu na slici 19.3. U trenutku kada se vrsta u kojoj se nalazi pritisnuti taster postavi na logičku nulu, odgovarajuća kolona će takođe biti postavljena na logičku nulu, usled kratkog spoja. Ova promena će biti registrovana na odgovarajućem pinu mikrokontrolera, na osnovu čega će biti poznato koji taster je pritisnut.



Slika 19.3: Ilustracija skeniranja matrične tastature

19.1.1 Biblioteka za rad sa matričnom tastaturom

U nastavku je dato objašnjenje svih funkcija unutar biblioteke **keypad**, koje se koriste za rad sa matričnom tastaturom. Biblioteka je dostupna za preuzimanje na GIT repozitorijumu ove knjige, putem sledećeg linka.

<https://github.com/rszes/biblioteke.git>

```
void keypadInit(uint8_t *rows, uint8_t *cols);
```

Opis: Funkcija koja vrši inicijalizaciju matrične tastature, u skladu sa prosleđenim parametrima **rows** i **cols**.

Parametri:

- **rows** – celobrojni niz koji sadrži brojeve digitalnih ulaza Arduino ploče koji predstavljaju vrste matrične tastature; prilikom navođenja datih pinova, njihov redosled treba biti isti kao i prilikom povezivanja periferije
- **cols** – celobrojni niz koji sadrži brojeve digitalnih ulaza Arduino ploče koji predstavljaju kolone matrične tastature; prilikom navođenja datih pinova, njihov redosled treba biti isti kao i prilikom povezivanja periferije

Povratna vrednost:

Nema.

```
int8_t keypadScan();
```

Opis: Funkcija koja vrši skeniranje trenutnog stanja matrične tastature po objašnjrenom principu, kolonu po kolonu.

Parametri:

Nema.

Povratna vrednost:

ASCII kod koji predstavlja pritisnuti taster, ili vrednost 0 ukoliko niti jedan od tastera nije pritisnut.

Primer programa koji demonstrira rad sa matričnom tastaturom

Primer 19.1.1. Napisati program koji očitava stanje matrične tastature i ispisuje trenutno pritisnuti taster putem serijskog terminala.

Rešenje:

Na početku, potrebno je definisati nizove koji predstavljaju vrste i kolone. Kao vrste se koriste niža četiri bita porta B, koji odgovaraju digitalnim ulazima 11, 10, 9 i 8 na Arduino Uno ploči. Sa druge strane, viša četiri bita porta D, odnosno digitalni ulazi 7, 6, 5 i 4, predstavljaju kolone. U okviru *main* funkcije, potrebno je izvršiti poziv inicijalizacionih funkcija za serijsku komunikaciju i rad sa matričnom tastaturom. Unutar beskonačne petlje, vrši se skeniranje matrične tastature i ispis trenutno pritisnutog tastera putem serijskog terminala. Obratiti pažnju da softversko diferenciranje nije realizovano u okviru postojećeg rešenja.

Rešenje primera je dato na listingu 19.1.

```
#include <avr/io.h>
#include <stdint.h>
#include "uart.h"
#include "keypad.h"

// niza cetiri pina porta B, odnosno digitalni ulazi 11,10,9 i
// 8
uint8_t rows[4] = {11, 10, 9, 8};
// visa cetiri pina porta D, odnosno digitalni ulazi 7,6,5 i 4
uint8_t cols[4] = {7, 6, 5, 4};

int16_t main()
{
    uint8_t button;
    // inicijalizacija serijske komunikacije
    usartInit(9600);
    // inicijalizacija maticne tastature
    keypadInit(rows, cols);

    while(1)
    {
        // ocitavanje tastera
        button = keypadScan();
        if (button != 0x00)
        {
            usartPutChar(button);
            usartPutChar('\n');
        }
    }

    return 0;
}
```

Listing 19.1: Rešenje Primera 19.1.1

19.2 Zadaci za vežbu

Zadatak 19.2.1. Napisati program koji detektuje pritisak tastera i šalje indikaciju pritisnutog tastera na serijski terminal. Pri tome, važi pravilo jedan pritisak – jedan karakter, bez obzira na to koliko dugo korisnik drži taster u pritisnutom stanju.

Zadatak 19.2.2. Napisati program koji proverava da li je broj prost i ispisuje ga u binarnom, decimalnom i heksadecimalnom formatu putem serijskog terminala. Unos broja se vrši putem matrične tastature od strane korisnika. Nakon unetog broja, pritiskom na taster #, traženi parametri se ispisuju. Primer unosa je dat u nastavku.

ULAZ (matrična tastatura): 31

IZLAZ (serijski terminal): prost, bin:11111, dec:31, hex:1F

Zadatak 19.2.3. Napisati program koji simulira jednostavan kalkulator na sledeći način. Korisnik zadaje dva decimalna broja i operator između njih preko matrične tastature. Nakon unosa prvog broja pritiskom tastera koji predstavljaju decimalne cifre, zadaje se operator pritiskom na neki od tastera A, B, C, ili D (A = '+', B = '-', C = '*', D = '/'), nakon čega se unosi drugi broj. Po završetku unosa izraza, pritiskom na taster * izračunava se vrednost izraza i ispisuje preko serijskog terminala. Primer unosa je dat u nastavku.

ULAZ (matrična tastatura): 57A12*

IZLAZ (serijski terminal): 57+12=69

Zadatak 19.2.4. Napisati program koji omogućava korisniku da odredi rešenja kvadratne jednačine. Koeficijenti kvadratne jednačine a , b i c se unose putem matrične tastature, nakon čega se pritiskom na tastera A, B ili C određuje koeficijent čija je vrednost unesena. Nakon pritiska tastera D se prikazuju rešenja kvadratne jednačine putem serijskog porta. Da bi se rešenja prikazala, potrebno je uneti sva tri koeficijenta. Pritiskom na taster # moguće je promeniti znak koeficijentu. Primer unosa je dat u nastavku.

ULAZ (matrična tastatura): 1A2#B3CD

IZLAZ (serijski terminal): a=1 b=-2 c=3

IZLAZ (serijski terminal): x1=1.00+i*1.41 x2=1.00-i*1.41

Zadatak 19.2.5. Napisati program koji proverava da li je korisnik ljudsko biće. Na početku, potrebno je generisati slučajan niz karaktera dužine od 5 do 9 karaktera. Podrazumevati da karakteri mogu biti brojevi 0-9 i slova A-D. Dužinu niza, takođe, generisati na slučajan način. Tako generisan niz karaktera ispisati putem serijskog terminala. Nakon toga, potrebno je preko matrične tastature uneti prikazani niz karaktera. Ukoliko je niz ispravno unet, putem serijskog terminala potrebno je ispisati "Ispravno unet kod", odnosno "Neispravno unet kod" ukoliko niz nije ispravno unet.

Zadatak 19.2.6. Napisati program koji simulira proveru PIN koda korisnika na bankomatu.

- Realizovati unos imena i prezimena korisnika putem serijskog terminala, a potom izvršiti proveru da li se korisnik nalazi u bazi. Primer jednostavne baze korisnika je dat u nastavku.

```
#define NUM_OF_USERS 5

int8_t user_database[NUM_OF_USERS][32] = {
    "Marko\u0107Markovic",
    "Milos\u0107Milosevic",
    "Petar\u0107Petrovic",
    "Jovan\u0107Jovanovic",
    "Janko\u0107Jankovic"
};
```

- Ukoliko se korisnik nalazi u bazi, omogućiti unos PIN koda putem matrične tastature. PIN kod predstavlja četvorocifreni decimalni broj koji se unosi karakter po karakter. Prilikom unosa pina potrebno je izvršiti ispis '*' preko serijskog terminala za svaki uneti karakter. Unos se okončava pritiskom na taster '#'. Po okončanju unosa, obavestiti korisnika da li je PIN ispravno unet ili ne. Primer baze korisničkih PIN-ova je dat u nastavku.

```
int8_t users_pin_database[NUM_OF_USERS][32] = {
    "3595",
    "7897",
    "5656",
    "3373",
    "5114"
};
```

Zadatak 19.2.7. Modifikovati Zadatak 19.2.5 tako da generisani niz, ovog puta, može uzimati vrednosti brojeva 0-9 i slova *A-Z* (dakle, sva slova Abecede). Pri tome, potrebno je omogućiti unos svih slova putem matrične tastature, na sledeći način:

- cifra 2 koduje broj 2 i slova a, b i c
- cifra 3 koduje broj 3 i slova d, e i f
- cifra 4 koduje broj 4 i slova g, h i i
- cifra 5 koduje broj 5 i slova j, k i l
- cifra 6 koduje broj 6 i slova m, n i o
- cifra 7 koduje broj 7 i slova p, q, r i s
- cifra 8 koduje broj 8 i slova t, u i v

- cifra 9 koduje broj 9 i slova w, x, y i z

Ovo je ilustrovano i na sledećoj slici.



Slika 19.4: Ilustracija rasporeda cifara i karaktera koje one koduju

Ukoliko je vremenski interval između dva pritiska istog tastera manji od 2s, omogućiti promenu ispisa karaktera na trenutnoj poziciji kurzora. U suprotnom, ukoliko je interval između dva pritiska istog tastera veći od 2s, zadržava se poslednji izabran karakter i kurzor se pomera za jedno mesto u desno (*opisani postupak unosa karaktera je vrlo sličan onom kod starijih verzija mobilnih telefona sa tastaturom*). Na primer, prilikom tri pritiska tastera 6, pri čemu je između svaka dva pritiska tastera vremenski interval bio manji od 2s, na izlazu će se ispisati karakter n (prvi pritisak tastera - broj 6, drugi pritisak tastera - slovo m i treći pritisak tastera - slovo n). Prilikom unosa putem matrične tastature, vrednosti ispisivati putem serijskog terminala.

Takođe, na sličan način kao i u Zadataku 19.2.5, potrebno je proveriti da li je niz ispravno unet i ukoliko jeste, putem serijskog terminala potrebno je ispisati "Ispravno unet kod". U suprotnom, potrebno je ispisati "Neispravno unet kod".

Zadatak 19.2.8. Napisati program koji implementira igru iks-oks (eng. *tic-tac-toe*) za dva igrača. Matrična tastatura predstavlja tablu za igru, gde brojevi 1-9 predstavljaju validna polja (odnosno, formiraju mrežu 3×3). Igrači naizmenično pritiskaju tastere, a pobeduje onaj igrač koji ostvari tri simbola u nizu (vertikalno, horizontalno ili dijagonalno). Trenutne vrednosti polja na tabli za igru ispisivati putem serijskog terminala. Inicijalno, polja na tabli su prazna. Konačno, ukoliko je pobedio neki od igrača ili je igra nerešena, putem serijskog terminala ispisati odgovarajuću poruku.

Zadatak 19.2.9. Napisati program koji implementira igru sličnu popularnoj igri *Guitar Hero*. Putem serijskog terminala se ispisuju karakteri brojeva 0-9 i slova A-D jedan po jedan, generisani na slučajan način. Korisnik putem matrične tastature unosi odgovarajuće vrednosti karaktera koje su prikazane na serijskom terminalu.

Pri tome, potrebno je da korisnik unese trenutno prikazani karakter u odgovarajućem vremenskom intervalu. Inicijalno, ovaj vremenski interval iznosi 2s, pri čemu se umanjuje za 100ms nakon svakih 10 uspešnih unosa. Za svaki uspešan unos, ukupan broj pogodaka se uvećava. U suprotnom ukoliko je unos neispravan, prikazuje se novi slučajan karakter i korisnik gubi jedan, od ukupno tri života. Igra se završava kada korisnik izgubi sva tri života. Na kraju, potrebno je ispisati ukupan broj pogodaka u serijskom terminalu.