СРЕДЊА ШКОЛА Кандидат: Зубац Андреја

"17. септембар" Струка: електротехника

Лајковац Смер: електротехничар рачунара

Школска: 2016/2017

# МАТУРСКИ РАД

## Тема: Рад са микроконтролерима у C-у – противпровална централа

**Напомена:** Задатак израђује кандидат самостално. У току израде кандидат се може према договору консултовати са наставником од кога је добио задатак. Рад треба да буде илустрован цртежима, скицама и шемама. Рад за проверу оспособљености се полаже пред комисијом. Завршни испит мора бити предат у 3 примерка од којих се два предају школи.

Примљено Теуму издао наставник:

\_\_\_\_\_\_\_\_\_\_\_

Саша Николић

Кандидат:

Зубац Андреја

# Садржај

1. Увод . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2. Противпровална централа . . . . . . . . . . . . . . . . . . . . . . . . . . 3

3. Опис програма . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

3.1 KontrolerRada . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

3.2 UnosSaTastature . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

3.3 CekajOtpustTastera . . . . . . . . . . . . . . . . . . . . . . . . . . 6

3.4 interrupt . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

3.5 IspisiLogo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

3.6 PromeniUlaznoIzlaznoVreme . . . . . . . . . . . . . . . . . . 6

3.7 UkljuciSigurnosniMod . . . . . . . . . . . . . . . . . . . . . . . 6

3.8 Resetuj . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

4. Програм . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

5. Закључак . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

# 1. УВОД

Рад са микроконтролерима је у данашње време много битан јер су они свуда око нас: рачунари, аутомобили, калкулатор, телефони и др... Микроконтролер је лакше речено мали рачунар који може да управља уређајима који су повезани на њега, а да би се то остварило он мора да садржи следеће:

* Микропроцесор
* Оперативну меморију (RAM меморија)
* Дигиталне и аналогне улазе и излазе
* Аналогно-дигитални конвертор (А/Д конвертор)
* Осцилатор
* Комуникационе склопове
* Програмску меморију

Дизајнирани су да раде у реалном времену (real-time control), нису скупи, мала је потрошња струје, масовно се производе и отпорни су на варијације напона, температуре, влажности, вибрације и тако даље. Могу се програмирати у вишим програмским језицима као што су:

Basic, Pascal и C.

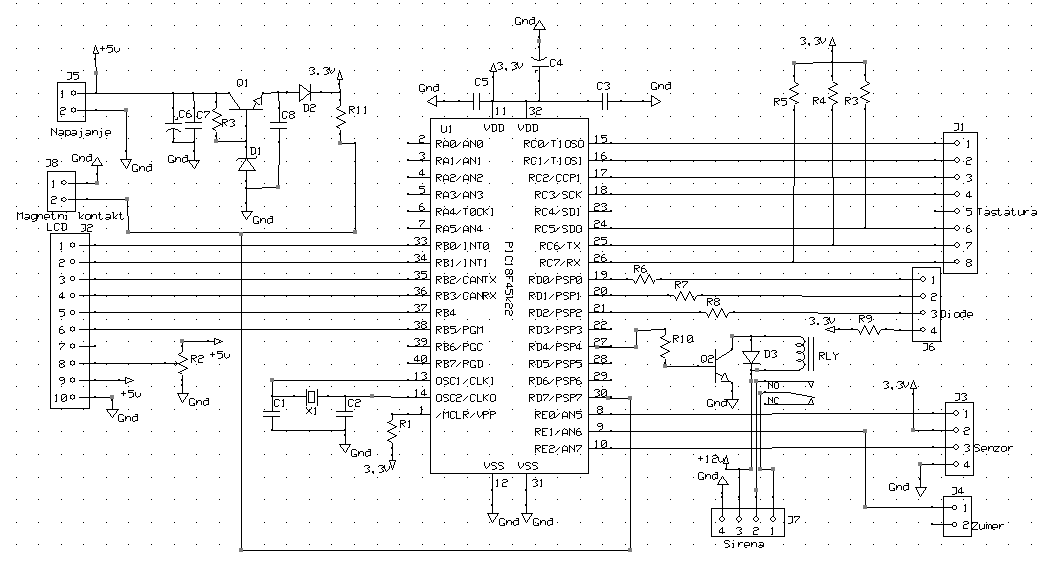
# 2. ПРОТИВПРОВАЛНА ЦЕНТРАЛА

Противпровална централа урађена је у C програмском језику помоћу mikroC компајлера који је направила фирма MikroElektronika.

Састоји се од:

* Напајања
* Тастатуре
* LCD дисплеја
* Диода
* Сирене
* Магнетног контакта
* Зумера
* Екстерног осцилатора
* Микроконтролера PIC18F45K22

На следећој слици је приказана шема противпровалне централе:



Слика 1

Постоје две алармне зоне, једна окида на сензор а друга на магнетни контакт који се налази на вратима. Шифра је четвороцифрена и можемо је мењати, у случају да заборавимо шифру постоји фабричка шифра која ресетује цео систем. Постоји улазно-излазно време које се активира када ми улазимо у просторију и када је напуштамо, тј. када укључујемо и искључујемо сигурносни мод. Сигурносни мод се активира када протекне излазно време након уноса исправне шифре. Када је централа у сигурносном моду она проверава помоћу сензора да ли је неко упао у просторију а да није отворио врата, и ако јесте активира се аларм без улазног времена. При активацији аларма пале се сирена и упозоравајуће светло, а деактивација аларма се врши уносом исправне шифре. У случају када отворимо врата, магнетни контакт детектује то и активира се улазно време које је променљиво у програмском моду и ако унесемо исправну шифру за то време излазимо из сигурносног мода и централа се деактивира. Ако не унесемо исправну шифру за то време онда се активира аларм који се искључује исправном шифром. Централа има и програмски мод у коме се могу мењати шифра и улазно-излазно време. У програмски мод улазимо када укуцамо исправну шифру и затим тарабу, а у сигурносни мод када укуцамо исправну шифру и звездицу. Када хоћемо да ресетујемо централу онда унесемо \*9753#, систем ће се ресетовати и шифра ће бити 9876 а улазно-излазно време 15 секунди.

# 3. ОПИС ПРОГРАМА

Програм се састоји од следећих функција:

* 3.1 void KontrolerRada();
* 3.2 void UnosSaTastature();
* 3.3 void CekajOtpustTastera();
* 3.4 void interrupt();
* 3.5 void IspisiLogo();
* 3.6 void PromeniUlaznoIzlaznoVreme(short);
* 3.7 void UkljuciSigurnosniMod();
* 3.8 void Resetuj();

## 3.1 KontrolerRada

Ово је функција које регулише рад противпровалне централе, у њој се позивају све остале функције по потреби. У главној функцији main се позива ова функција и извршава се непрекидно јер се налази и петљи како би микроконтролер радио без прекида.

На почетку ове функције позивамо функцију UnosSaTastature() која испитује да ли је притиснут неки тастер. Затим испитујемо да ли променљива trenutnoPritisnutiTaster није једнака броју 13 и ако је тај услов испуњен то значи да смо притиснули неки тастер.

Када је притиснут неки тастер активира се зумер позивом функције

Sound\_Play(фреквенција, трајање), а ако притиснемо тастер кад нисмо у сигурносном моду и не урадимо ништа у току 10 секунди, централа се враћа у претходно стање и исписује лого на дисплеју. Након сваког притиска тастера позива се функција CekajOtpustTastera(). Кад је лого на екрану и ако притиснемо било који тастер појавиће се мени, под бројем један је улаз за сигурносни, програмски мод а и могућност ресетовања централе. Кад унесемо четвороцифрену шифру након притиска звездице испитује се да ли је шифра исправна, пореди се са исправном шифром која се налази у EEPROM меморији (меморија која се чува након нестанка струје и може се програмски брисати). Ако је исправна улазимо у сигурносни мод а ако није онда се исписује да шифра није исправна и затим лого. У програмски мод улазимо са исправном шифром и тарабом, постоје две опције: промена улазно-излазног времена и промена шифре. Промена улазно-излазног времена се тастерима 1 и 2 смањује и повећава а звездицом се чува у меморији. Промена шифре, уносимо нову шифру и притиском звездице се чува у меморији. Ресетовање система се врши комбинацијом \*9753# и то траје око 2 до 3 секунде.

## 3.2 UnosSaTastature

Ова функција скенира унос са тастатуре на следећи начин.

Имамо тастатуру 4х3 (4 реда и 3 колоне) и на редове шаљемо логичку `0` па испитујемо где се појавила логичка нула када притиснемо тастер. На пример у коду смо послали на први ред `0` и испитујемо колоне за тај ред да ли се појавила `1`, она се појављује када ми притиснемо тастер (доводимо напон са микроконтролера) и ако је први ред притиснут у првој колони онда је то тастер 1, ако је други ред притиснут онда је тастер 2 и тако даље. У случају да ништа не притиснемо променљива trenutnoPritisnutiTaster ће имати вредност 13 што ми очитавамо у програму као да ништа није притиснуто.

## 3.3 CekajOtpustTastera

Ова функција нам је потребна јер када ми физички притиснемо тастер он се неће идеално додирнути једном за проводник и провести струју. Он ће у реалном случају више пута додирнути проводник и провести што није добро и то се зове одскакање тастера (dobunce ефекат). Ми то спречавамо софтверски тако што у регистре микроконтролера уносимо `0` или `1` и када је тастер притиснут вредности ће бити различите, када пустимо код ће прећи на следећу линију кода. Ово је урађено јер смо хтели да микроконтролер региструје који је тастер притиснут тек након његовог отпуштања.

## 3.4 interrupt

У овој функцији испитујемо када се десио догађај (прекид), овај програм користи два тајмера и овде региструјемо када они одброје задато време. Ово нам служи за улазно-излазно време и исписивање логоа након неког времена када се притисне било који тастер. Interrupt је функција која се увек извршава поред главне функције main и много је важна за напредно програмирање микроконтролера.

## 3.5 IspisiLogo

Ова функција исписује на LCD-u лого наше школе помоћу једноставних функција.

Функција Lcd\_Out (ред, колона, текст) служи за испис.

Функција Lcd\_Cmd (команда) извршава задату команду.

## 3.6 PromeniUlaznoIzlaznoVreme

Ова функција има један параметар који нам казује да ли је инкрементирање или декрементирање времена (+1 или -1). Овде прво вршимо додељивање вредности променљивој temp и затим је исписујемо на дисплеју. Пошто је temp типа int морамо је конвертовати у текст па је тек онда исписати. Такође се не може декрементирати ако је вредност 0.

## 3.7 UkljuciSigurnosniMod

У овој функцији активирамо тајмер и кад год тајмер одброји у interrupt функцији се то региструје па у главној функцији када одбројавање буде једнако излазном времену активира се сигурносни мод.

## 3.8 Resetuj

Комбинацијом \*9753# се позива ова функција која ресетује целу централу. Улазно-излазно време поставља на 15 секунди (у EEPROM меморију уписујемо вредности) и шифру на 9876.

# 4. ПРОГРАМ

Код програма је следећи:

sbit LCD\_RS at LATB4\_bit;

sbit LCD\_EN at LATB5\_bit;

sbit LCD\_D4 at LATB0\_bit;

sbit LCD\_D5 at LATB1\_bit;

sbit LCD\_D6 at LATB2\_bit;

sbit LCD\_D7 at LATB3\_bit;

sbit LCD\_RS\_Direction at TRISB4\_bit;

sbit LCD\_EN\_Direction at TRISB5\_bit;

sbit LCD\_D4\_Direction at TRISB0\_bit;

sbit LCD\_D5\_Direction at TRISB1\_bit;

sbit LCD\_D6\_Direction at TRISB2\_bit;

sbit LCD\_D7\_Direction at TRISB3\_bit;

void KontrolerRada();

void UnosSaTastature();

void CekajOtpustTastera();

void interrupt();

void IspisiLogo();

void PromeniUlaznoIzlaznoVreme(short);

void IzracunajVreme();

void IspisiVreme();

void UkljuciSigurnosniMod();

void Resetuj();

short trenutnoPritisnutiTaster, i, unosBrojac = 0;

short meniStanje = 0, postojiLiSifra = 0, pokreniVreme = 0, prvaZvezdica = 0, programskiMod = 0, detektovan = 0, unesiteSifru = 0,

sigurnosniMod = 0, alarmUkljucen = 0, promenaSifre = 0, promenaUlIzVremena = 0, prvaCifra = 0, postojiLiUlIzVreme = 0,

ulazIzlazTajmer = 0, ampm = 1, daLiJeVremePodeseno = 0, podesitiVreme = 0, mozePrikazatiVreme = 0, napoljuUnutra = 0,

uSigurnosnomModu = 0;

int sifraUnos[4], sifraCitanje[4], sifraProcitana, sifraUneta, ulaznoIzlaznoVreme = 10, tajmer1Brojac = 0, tajmer0Brojac = 0, temp;

int sat = 100, minut = 100, sekund = 100;

char ispisiCifru, ispisiUlIzVreme[4], \_sat[7], \_minut[7], \_sekund[7], ispisiTacnoVreme[7];

void main() {

ANSELA = ANSELC = ANSELB = ANSELE = ANSELD = 0;

TRISC = 0b11100000;

TRISB = 0x00;

TRISE = 0x01;

TRISD = 0x00;

TRISD.B7 = 1;

LATD = 0;

LATD.B2 = 1;

T1CON = 0x30;

TMR1IF\_bit = 0;

TMR1H = 0x00;

TMR1L = 0x00;

TMR1IE\_bit = 1;

INTCON.PEIE = 1;

INTCON.GIE = 1;

T0CON.T08BIT = 0;

T0CON.PSA = 0;

T0CON.T0PS0 = 1;

T0CON.T0PS1 = 0;

T0CON.T0PS2 = 1;

TMR0H = 0x85;

TMR0L = 0xEF;

T0CON.T0CS = 0;

TMR0IF\_bit = 0;

TMR0ON\_bit = 0;

TMR0IF\_bit = 1;

TMR0IE\_bit = 1;

ulazIzlazTajmer = 0;

Sound\_Init(&PORTE, 1);

Lcd\_Init();

LATE.B2 = 0;

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(30);

Lcd\_Out(1, 1, " UCITAVANJE ");

Lcd\_Out(2, 1, " SISTEMA! ");

postojiLiSifra = EEPROM\_Read(0x00 + 4);

Delay\_ms(30);

postojiLiUlIzVreme = EEPROM\_Read(0x00 + 5);

Delay\_ms(30);

daLiJeVremePodeseno = EEPROM\_Read(0x00 + 7);

Delay\_ms(30);

uSigurnosnomModu = EEPROM\_Read(0x00 + 8);

Delay\_ms(30);

if (uSigurnosnomModu == 1)

{

UkljuciSigurnosniMod();

}

if (postojiLiUlIzVreme != 1)

{

EEPROM\_Write(0x00 + 6, 15);

Delay\_ms(30);

EEPROM\_Write(0x00 + 5, 1);

Delay\_ms(30);

ulaznoIzlaznoVreme = 15;

}

else

{

ulaznoIzlaznoVreme = EEPROM\_Read(0x00 + 6);

Delay\_ms(30);

}

if (postojiLiSifra != 1)

{

EEPROM\_Write(0x00 + 0, 9);

Delay\_ms(30);

EEPROM\_Write(0x00 + 1, 8);

Delay\_ms(30);

EEPROM\_Write(0x00 + 2, 7);

Delay\_ms(30);

EEPROM\_Write(0x00 + 3, 6);

Delay\_ms(30);

EEPROM\_Write(0x00 + 4, 1);

Delay\_ms(30);

}

Delay\_ms(2000);

if (uSigurnosnomModu != 1)

IspisiLogo();

while (1)

{

if (PORTD.B7 == 1 && sigurnosniMod == 1 && alarmUkljucen == 0 && detektovan == 0)

{

detektovan = 1;

tajmer0Brojac = 0;

ulazIzlazTajmer = 1;

TMR0ON\_bit = 1;

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, " ULAZNO VREME ");

Lcd\_Cmd(\_LCD\_SECOND\_ROW);

unesiteSifru = 1;

unosBrojac = 0;

}

if (PORTE.B0 == 1 && alarmUkljucen == 0 && sigurnosniMod == 1)

{

LATD.B0 = 1;

LATD.B4 = 1;

alarmUkljucen = 1;

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, "ALARM ZONA 2");

Lcd\_Cmd(\_LCD\_SECOND\_ROW);

Sound\_Play(1000, 5000);

unesiteSifru = 1;

unosBrojac = 0;

TMR0ON\_bit = 0;

tajmer0Brojac = 0;

}

if (tajmer1Brojac >= 30)

{

if (sigurnosniMod == 1)

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, "Vasa sifra->");

Delay\_ms(50);

unesiteSifru = 1;

unosBrojac = 0;

tajmer1Brojac = 0;

TMR1ON\_bit = 0;

}

else

{

IspisiLogo();

meniStanje = 0;

unesiteSifru = 0;

unosBrojac = 0;

tajmer1Brojac = 0;

TMR1ON\_bit = 0;

}

}

KontrolerRada();

if (tajmer0Brojac >= ulaznoIzlaznoVreme) // && ulazIzlazTajmer == 1)

{

if (sigurnosniMod == 1)

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, "ALARM ZONA 1");

Lcd\_Cmd(\_LCD\_SECOND\_ROW);

Sound\_Play(1000, 5000);

unosBrojac = 0;

alarmUkljucen = 1;

LATD.B4 = 1;

TMR0ON\_bit = 0;

tajmer0Brojac = 0;

LATD.B1 = 1;

}

else

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, " SIGURNOSNI MOD ");

Delay\_ms(1000);

sigurnosniMod = 1;

alarmUkljucen = 0;

tajmer0Brojac = 0;

unosBrojac = 0;

napoljuUnutra = 0;

//ulazIzlazTajmer = 0;

TMR0ON\_bit = 0;

}

}

}

}

void KontrolerRada()

{

UnosSaTastature();

if (trenutnoPritisnutiTaster != 13)

{

Sound\_Play(1000, 60);

if (detektovan == 0 && programskiMod == 0 && podesitiVreme == 0)

{

TMR1H = 0x00;

TMR1L = 0x00;

TMR1ON\_bit = 1;

}

if (meniStanje == 0 && unesiteSifru == 0 && sigurnosniMod == 0 && programskiMod == 0 && promenaUlIzVremena == 0)//sigurnosniMod == 0 && unosBrojac == 0 && postaviSifru == 0 && unesiSifru == 0 && programskiMod == 0)

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

Lcd\_Out(1, 1, "1.Sigurnosni mod");

meniStanje = 1;

unosBrojac = 1;

CekajOtpustTastera();

return;

}

if (meniStanje == 1)

{

if (trenutnoPritisnutiTaster == 1)

{

if (promenaUlIzVremena == 1)

{

PromeniUlaznoIzlaznoVreme(-1);

return;

}

if (programskiMod != 1)

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, "Vasa sifra->");

}

else

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, "Nova sifra->");

promenaSifre = 1;

}

unesiteSifru = 1;

unosBrojac = 0;

meniStanje = 0;

CekajOtpustTastera();

return;

}

if (trenutnoPritisnutiTaster == 2)

{

if (promenaUlIzVremena == 1)

{

PromeniUlaznoIzlaznoVreme(1);

return;

}

meniStanje = 0;

if (programskiMod == 1)

{

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Out(1, 1, "1-> -1s 2-> +1s");

Delay\_ms(1500);

if (postojiLiUlIzVreme == 1)

{

ulaznoIzlaznoVreme = EEPROM\_Read(0x00 + 6);

Delay\_ms(30);

}

else

ulaznoIzlaznoVreme = 15;

temp = ulaznoIzlaznoVreme;

IntToStr(ulaznoIzlaznoVreme, ispisiTacnoVreme);

Lcd\_Out(2, 1, "Novo vreme->");

Lcd\_Out(2, 13, ispisiTacnoVreme + 2);

Delay\_ms(50);

meniStanje = 1;

}

unosBrojac = 0;

promenaUlIzVremena = 1;

//ulazIzlazTajmer = 0;

}

CekajOtpustTastera();

}

if (promenaUlIzVremena == 1)

{

if (trenutnoPritisnutiTaster == 10)

{

EEPROM\_Write(0x00 + 6, temp);

Delay\_ms(30);

if (postojiLiUlIzVreme != 1)

{

EEPROM\_Write(0x00 + 5, 1);

Delay\_ms(30);

}

ulaznoIzlaznoVreme = temp;

promenaUlIzVremena = 0;

programskiMod = 0;

unosBrojac = 0;

meniStanje = 0;

unesiteSifru = 0;

//ulazIzlazTajmer = 0;

CekajOtpustTastera();

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, " UL\IZ Vreme ");

Lcd\_Out(2, 1, " promenjeno ");

Delay\_ms(1500);

IspisiLogo();

}

}

if (unesiteSifru == 1)

{

if (trenutnoPritisnutiTaster != 10 && trenutnoPritisnutiTaster != 11 && unosBrojac < 4)

{

ShortToStr(trenutnoPritisnutiTaster, ispisiCifru);

Lcd\_Out\_CP(ispisiCifru + 3);

Delay\_ms(50);

sifraUnos[unosBrojac] = trenutnoPritisnutiTaster;

unosBrojac++;

tajmer1Brojac = 0;

CekajOtpustTastera();

return;

}

else if (unosBrojac == 0 && trenutnoPritisnutiTaster == 11 && prvaZvezdica == 0 && programskiMod == 0)

{

prvaZvezdica = 1;

unosBrojac = 0;

CekajOtpustTastera();

return;

}

if ((trenutnoPritisnutiTaster == 10 || trenutnoPritisnutiTaster == 11) && unosBrojac >= 3 && napoljuUnutra == 0)

{

if (promenaSifre == 1)

{

for (i = 0; i < 4; i++)

{

EEPROM\_Write(0x00 + i, sifraUnos[i]);

Delay\_ms(30);

}

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, " Sifra ");

Lcd\_Out(2, 1, " promenjena ");

Delay\_ms(1500);

meniStanje = 0;

promenaSifre = 0;

programskiMod = 0;

unosBrojac = 0;

unesiteSifru = 0;

//ulazIzlazTajmer = 0;

CekajOtpustTastera();

IspisiLogo();

return;

}

for (i = 0; i < 4; i++)

{

sifraCitanje[i] = EEPROM\_Read(0x00 + i);

Delay\_ms(30);

}

sifraProcitana = sifraCitanje[0] \* 1000 + sifraCitanje[1] \* 100 + sifraCitanje[2] \* 10 + sifraCitanje[3];

sifraUneta = sifraUnos[0] \* 1000 + sifraUnos[1] \* 100 + sifraUnos[2] \* 10 + sifraUnos[3];

if (sifraUneta == sifraProcitana)

{

promenaUlIzVremena = 0;

if (trenutnoPritisnutiTaster == 10)

{

if (sigurnosniMod == 1)

{

alarmUkljucen = 0;

detektovan = 0;

napoljuUnutra = 0;

unosBrojac = 0;

unesiteSifru = 0;

sigurnosniMod = 0;

tajmer0Brojac = tajmer1Brojac = 0;

TMR0ON\_bit = 0;

TMR1ON\_bit = 0;

prvaZvezdica = 0;

LATD.B0 = LATD.B1 = LATD.B4 = 0;

LATD.B2 = 1;

EEPROM\_Write(0x00 + 8, 0);

Delay\_ms(30);

CekajOtpustTastera();

IspisiLogo();

return;

}

UkljuciSigurnosniMod();

//sigurnosniMod = 1;

}

else

{

if (sigurnosniMod == 0)

{

for (i = 0; i < 3; i++)

{

Delay\_ms(250);

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(250);

Lcd\_Out(1, 1, " PROGRAMSKI MOD ");

Delay\_ms(250);

}

programskiMod = 1;

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Out(1, 1, "1.Nova sifra");

Lcd\_Out(2, 1, "2.UL/IZ vreme");

Delay\_ms(50);

ulazIzlazTajmer = 0;

prvaZvezdica = 0;

meniStanje = 1;

tajmer1Brojac = 0;

tajmer0Brojac = 0;

TMR1ON\_bit = 0;

}

}

CekajOtpustTastera();

}

else if (sifraUneta == 9753 && prvaZvezdica == 1 && trenutnoPritisnutiTaster == 10)

{

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Out(1, 1, " SISTEM ");

Lcd\_Out(2, 1, " RESETOVAN! ");

Sound\_Play(1000, 2000);

Delay\_ms(1000);

Resetuj();

meniStanje = 0;

unosBrojac = 0;

unesiteSifru = 0;

prvaZvezdica = 0;

unosBrojac = 0;

unesiteSifru = 0;

tajmer0Brojac = 0;

tajmer1Brojac = 0;

ulazIzlazTajmer = 0;

CekajOtpustTastera();

IspisiLogo();

}

else

{

if (sigurnosniMod == 1)

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, " Pogresna sifra ");

Delay\_ms(1000);

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, "Vasa sifra->");

unesiteSifru = 1;

unosBrojac = 0;

TMR0ON\_bit = 0;

tajmer0Brojac = 0;

CekajOtpustTastera();

return;

}

else

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Out(1, 1, " Pogresna sifra ");

Delay\_ms(1000);

unesiteSifru = 0;

unosBrojac = 0;

}

tajmer0Brojac = 0;

CekajOtpustTastera();

IspisiLogo();

}

}

}

}

}

void UnosSaTastature()

{

LATC = 0b11101110;

if (!PORTC.B7)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 1;

return;

}

if (!PORTC.B6)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 4;

return;

}

if (!PORTC.B5)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 7;

return;

}

LATC = 0b11101101;

if (!PORTC.B7)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 2;

return;

}

if (!PORTC.B6)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 5;

return;

}

if (!PORTC.B5)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 8;

return;

}

LATC = 0b11101011;

if (!PORTC.B7)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 3;

return;

}

if (!PORTC.B6)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 6;

return;

}

if (!PORTC.B5)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 9;

return;

}

LATC = 0b11100111;

if (!PORTC.B7)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 10; //Taraba

return;

}

if (!PORTC.B6)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 11; //Zvezda

return;

}

if (!PORTC.B5)

{

Delay\_ms(50);

trenutnoPritisnutiTaster = 0;

return;

}

trenutnoPritisnutiTaster = 13;

}

void CekajOtpustTastera()

{

PORTC = 0b11100000;

while(PORTC != 0b11100000)

PORTC = 0b11100000;

Delay\_ms(30);

}

void IspisiLogo()

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(50);

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

Lcd\_Out(1, 1, " SREDNJA SKOLA ");

Lcd\_Out(2, 1, " LAJKOVAC ");

Delay\_ms(50);

}

void interrupt()

{

if (TMR1IF\_bit == 1)

{

tajmer1Brojac++;

TMR1IF\_bit = 0;

}

if (TMR0IF\_bit == 1)

{

TMR0H = 0x85;

TMR0L = 0xEF;

TMR0IF\_bit = 0;

tajmer0Brojac++;

}

}

void PromeniUlaznoIzlaznoVreme(short plusMinus)

{

if (temp > 0 || (temp == 0 && trenutnoPritisnutiTaster == 2))

{

short duzinaStringa;

temp += plusMinus;

//Lcd\_Cmd(\_LCD\_CLEAR);

//Delay\_ms(50);

Lcd\_Out(1, 1, "1-> -1s 2-> +1s");

Lcd\_Cmd(\_LCD\_SECOND\_ROW);

Lcd\_Out(2, 1, "Novo vreme->");

IntToStr(temp, ispisiTacnoVreme);

Lcd\_Out(2, 13, ispisiTacnoVreme + 2);

Delay\_ms(20);

}

else

{

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(30);

Lcd\_Out(1, 1, " MORA BITI ");

Lcd\_Out(2, 1, " POZITIVNO ");

Delay\_ms(1500);

CekajOtpustTastera();

Lcd\_Out(1, 1, "1-> +1s 2-> -1s");

Lcd\_Cmd(\_LCD\_SECOND\_ROW);

Lcd\_Out(2, 1, "Novo vreme->");

Delay\_ms(20);

}

}

void UkljuciSigurnosniMod()

{

TMR1ON\_bit = 0;

TMR0ON\_bit = 1;

tajmer1Brojac = 0;

for (i = 0; i < 3; i++)

{

Delay\_ms(250);

Lcd\_Cmd(\_LCD\_CLEAR);

Delay\_ms(250);

Lcd\_Out(1, 1, " IZLAZNO VREME ");

Delay\_ms(250);

}

tajmer0Brojac = 0;

pokreniVreme = 1;

prvaZvezdica = 0;

napoljuUnutra = 1;

uSigurnosnomModu = 1;

LATD.B2 = 0;

}

void Resetuj()

{

EEPROM\_Write(0x00 + 6, 15);

Delay\_ms(30);

EEPROM\_Write(0x00 + 5, 1);

Delay\_ms(30);

EEPROM\_Write(0x00 + 0, 9);

Delay\_ms(30);

EEPROM\_Write(0x00 + 1, 8);

Delay\_ms(30);

EEPROM\_Write(0x00 + 2, 7);

Delay\_ms(30);

EEPROM\_Write(0x00 + 3, 6);

Delay\_ms(30);

EEPROM\_Write(0x00 + 4, 1);

Delay\_ms(30);

}

# 5. ЗАКЉУЧАК

Овај пример представља један од начина рада са микроконтролерима и како се то ради.

Можемо видети да програмирање микроконтролера није толико тешко као што сви мисле јер постоје алати који нам олакшавају посао. Са малим знањем програмирања (неким програмским језиком) може се прећи на учење како раде микроконтролери. Поред програмирања пожељно је знати и мало електронике како би имали више могућности за рад.

Као што видимо на микроконтролере можемо повезати доста модула са којима можемо радити као на пример: топлотни сензори, електромагнетни сензори, GSM (служи за слање порука, има га у свим телефонима) и тако даље...