

Računske vežbe iz  
Projektovanja Elektronskih  
Sistema  
cas 7

Doc.dr Borisav Jovanović

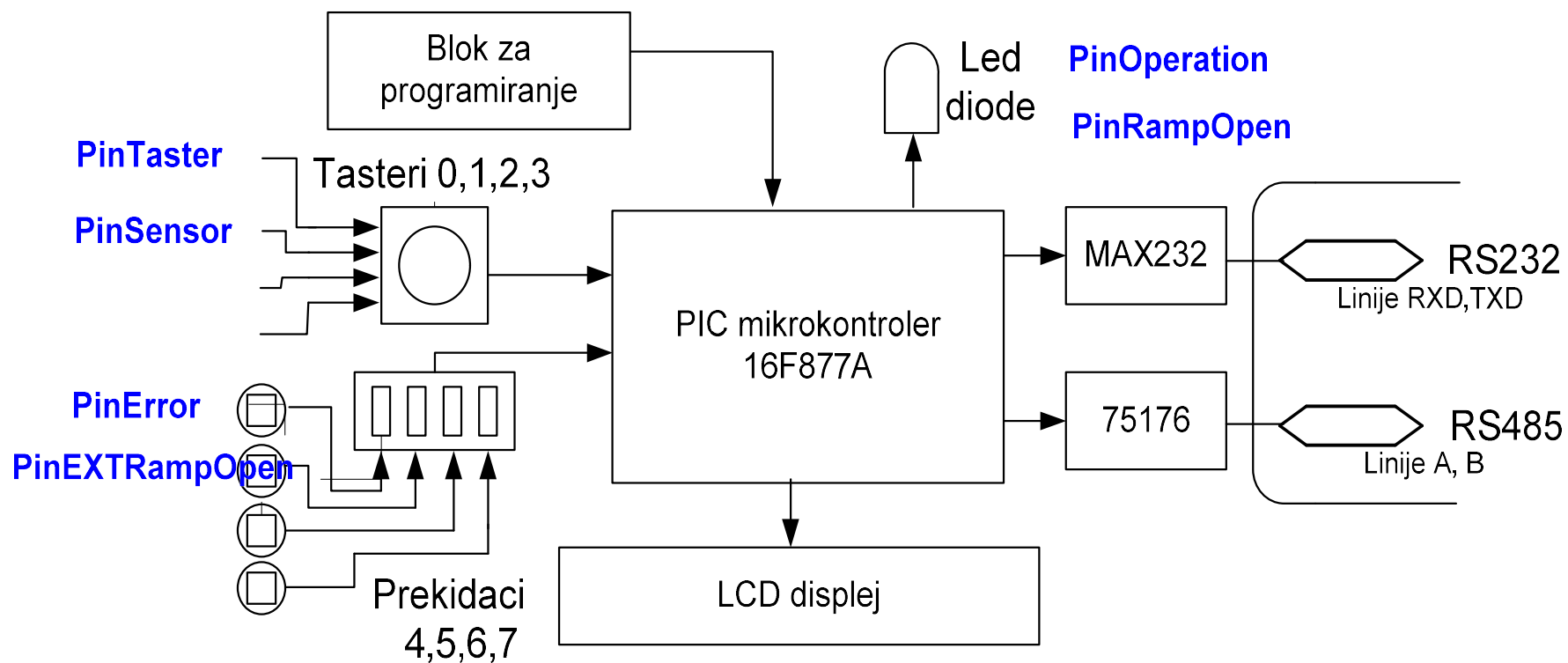
## Sadržaj:

- Realizacija hardvera Slejv automata “Podsistema ulaznih rampi”;
- Definisane osnovnog interfejsa (ulazi, izlazi);
- Detaljni opis hardvera. Šema konkretne štampane ploče koja se koristi za implementaciju Slejva.
- Tabela iskorišćenih ulaznih i izlaznih pinova mikrokontrolera
- Tabela korišćenih promenljivih u programu
- Tabela korišćenih funkcija u programu
- Programski kod mikrokontrolera
  - Definisane pinova i funkcija
  - Deklarisanje promenljivih
  - Funkcija **init\_variables()**
  - Funkcija **init()**
  - Funkcija **ProcessInputs()**
  - Funkcija **main()**

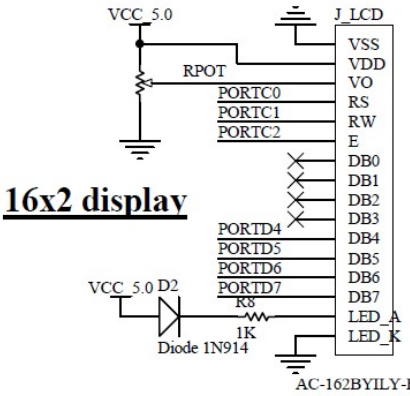
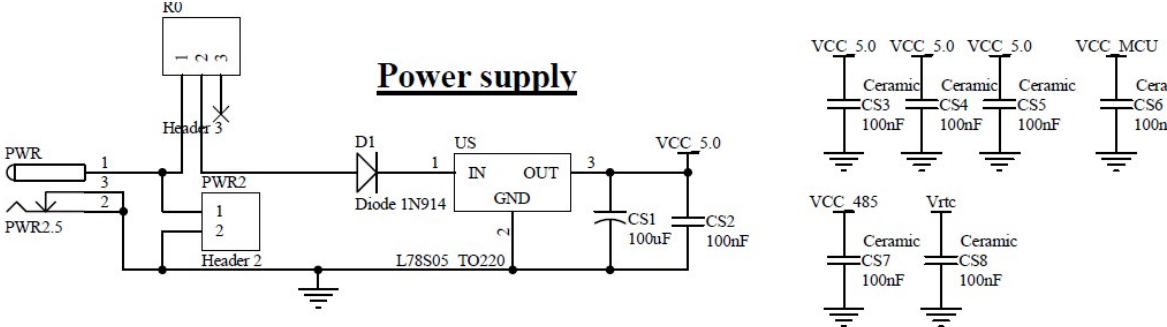


Realizacija hardvera Slejv automata  
“Podsistema ulaznih rampi” za  
PIC16F874A

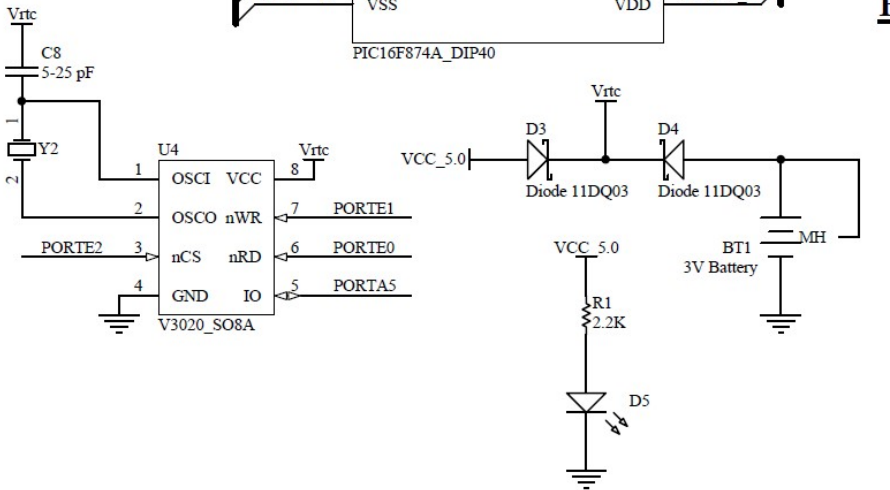
# Definisanje osnovnog interfejsa (ulazi, izlazi);

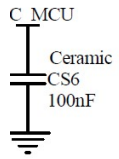


# Detailan opis hardvera

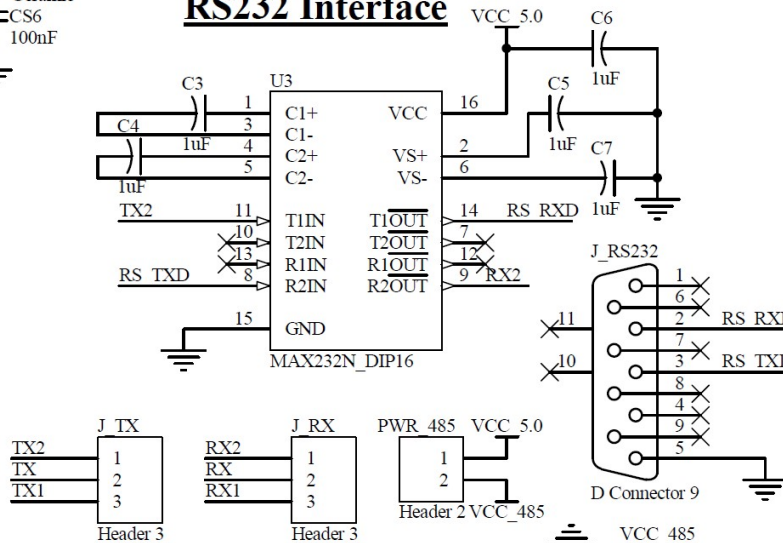


### PIC16F877A

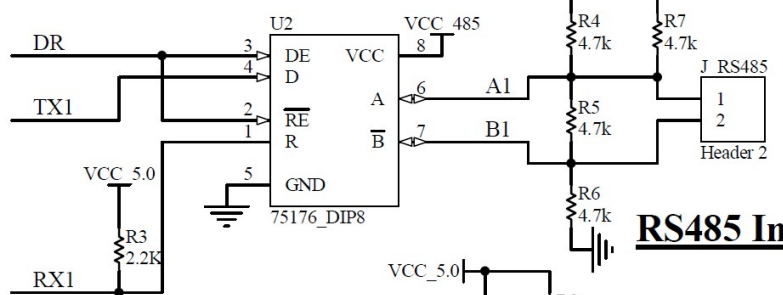
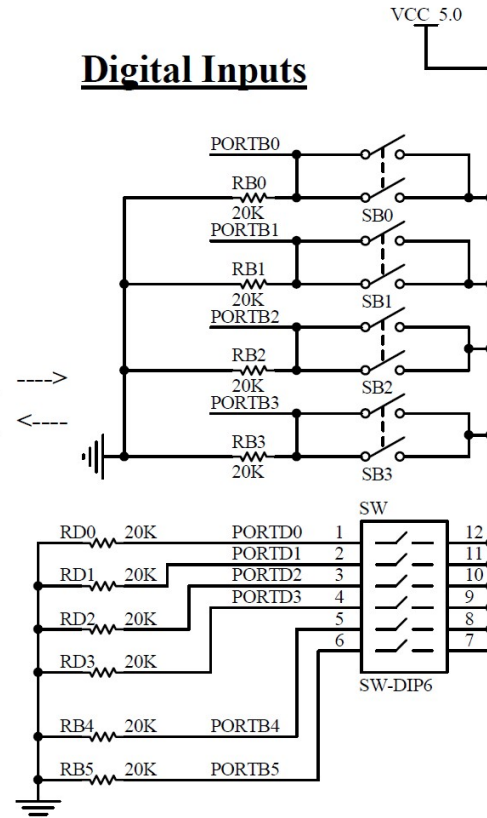




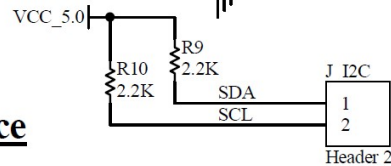
## RS232 Interface



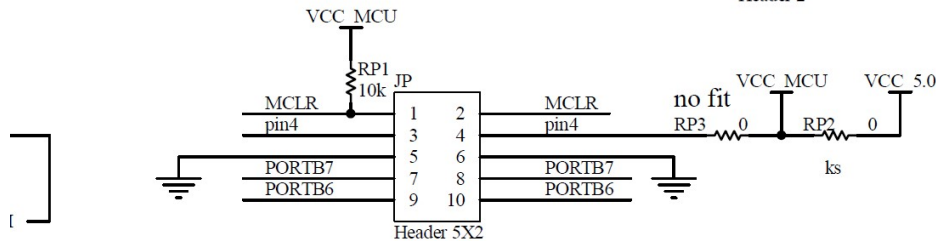
## Digital Inputs



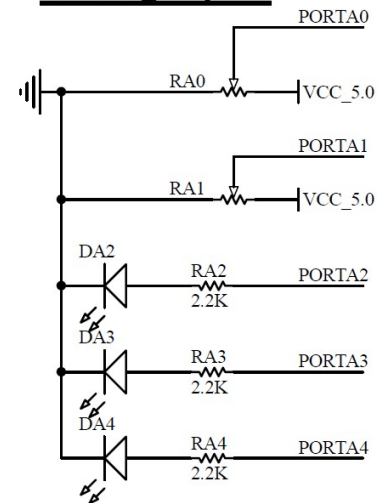
## RS485 Interface



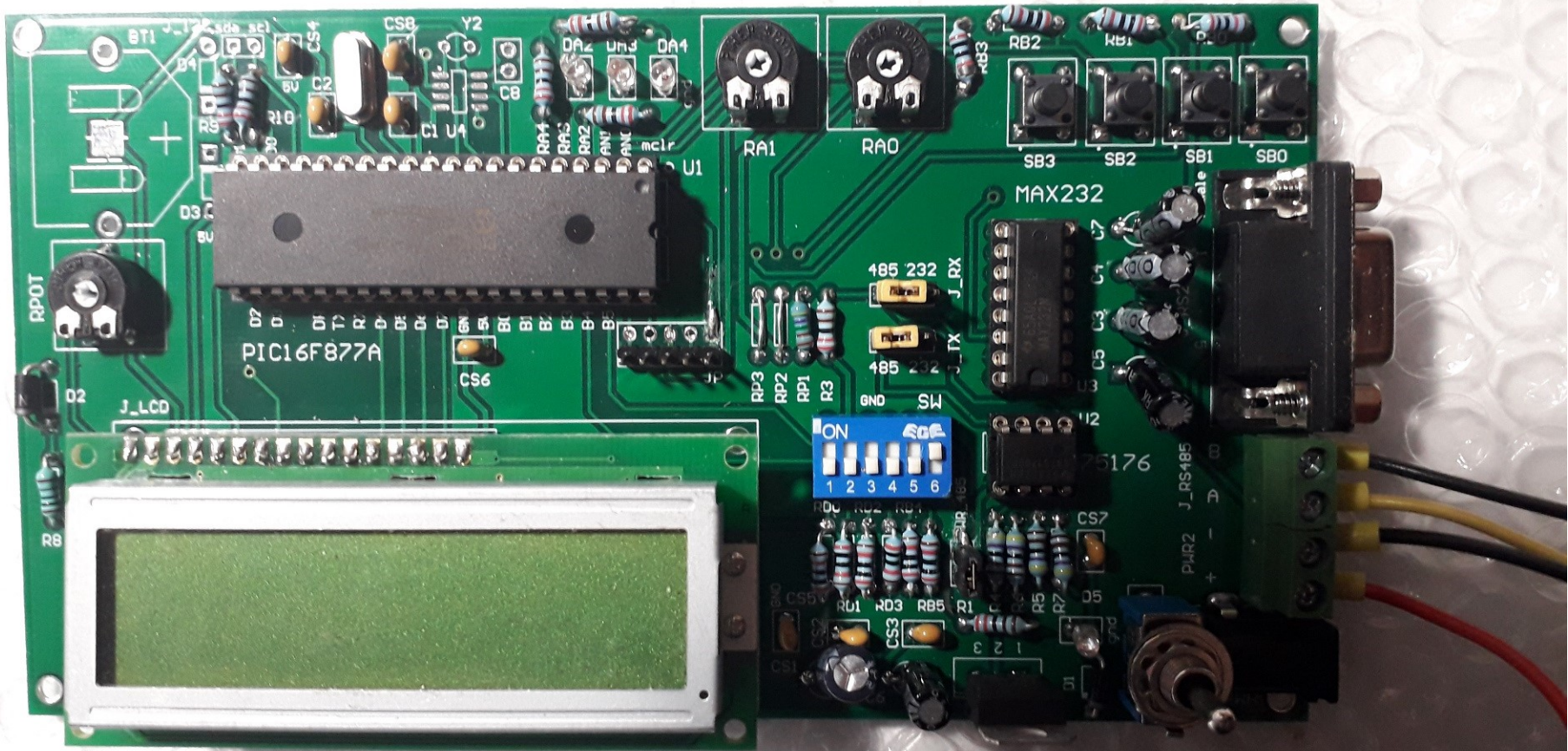
## Programmer Interface



## Analog Inputs











# Master

## Prozivka

KOMANDA		ID	RAMPE	Y $\in\{0,1\}$			
0	0	1	X	Y	Y	Y	Y

Podešavanje RTC. Posle ovog bajta, Slejv automat šalje još tri dodatna bajta: *sekunde, minuti, sati*

KOMANDA		ID	RAMPE	Y $\in\{0,1\}$			
0	1	1	X	Y	Y	Y	Y

# Slejv

Automat nema kartica.

KOMANDA		ID	RAMPE	Y $\in\{0,1\}$			
0	0	0	X	Y	Y	Y	Y

Automat radi ali nije pritisnut taster

KOMANDA		ID	RAMPE	Y $\in\{0,1\}$			
0	0	1	X	Y	Y	Y	Y

Pritisnut je taster, tj. prošlo je vozilo kroz rampu. Posle ovog bajta, Slejv automat šalje još četiri dodatna bajta: *sekunde, minuti, sati i kategorija vozila*.

KOMANDA		ID	RAMPE	Y $\in\{0,1\}$			
0	1	0	X	Y	Y	Y	Y

Odgovor da je RTC podešen

KOMANDA		ID	RAMPE	Y $\in\{0,1\}$			
0	1	1	X	Y	Y	Y	Y

- Za sate, minute, sekunde koje Master šalje Slejvovima
- $0x30+X$ , gde je X broj sati, minuta ili sekundi
- Na primer: ako se šalje 9 sati,  $0x30+0x09=0x39$ , ovo je ASCII kod za cifru 9
- ako se šalje 23 sati,  $0x30+0x17=0x47$ , ovo je ASCII kod za slovo 'G'.
- ako se šalje 59 minuta,  $0x30+0x3B=0x6B$ , ovo je ASCII kod za slovo 'k'.

## Raspored iskorišćenih ulaznih i izlaznih pinova mikrokontrolera

Port	Smer	Opis
RA2	izlaz	<b>PinOperation</b> - Led dioda. Simulira da je rampa u funkciji (svetli zeleno ili crveno svetlo)
RA3	izlaz	<b>PinRampOpen</b> - Led dioda. Simulira da je rampa otvorena
RB0	ulaz	<b>PinTaster</b> - taster - Zahtev vozača za preuzimanje kartice
RB1	ulaz	<b>PinSensor</b> – taster - Senzor prolaska vozila kroz rampu
RB4	ulaz	<b>PinError</b> – prekidač - Ako je RB4='1', simulira neispravan rad automata ili da kartice nedostaju
RB5	ulaz	<b>PinEXTRampOpen</b> – prekidač - Ako je RB5='1', rampa se otvara.
RD3, RD2	ulazi	Kategorija vozila
RC0	izlaz	<b>RS</b> - <i>Data/Instruction select</i> kontrolni signal LCD displeja
RC1	izlaz	<b>RW</b> - <i>Read/Write</i> kontrolni signal LCD displeja
RC2	izlaz	<b>E</b> - <i>Enable</i> kontrolni signal LCD displeja
RC5	izlaz	<b>DR</b> - <i>Write/Read</i> kontrola RS485 dražvera SN65176; ako je DR='1' kolo je u <i>write</i> modu rada, u suprotnom - <i>read</i> mod
RC6	izlaz	<b>TX</b> - <i>Transmit</i> signal UART-a mikrokontrolera
RC7	ulaz	<b>RX</b> - <i>Receive</i> signal UART-a
RD4 - RD7	izlaz	Magistrala podataka za prenos informacija između mikrokontrolera i LCD displeja.

## Raspored korišćenih promenljivih

Promenljive	Tip podataka	Objašnjenje promenljivih
Operation, Operation2	<i>bit</i>	Rampa u funkciji ako je Operation=1, Operation se programira kada Slejv dobije komadu za radi; Operation2 uzima u obzir stanje na Error pinu.
RampOpen, RampOpen2	<i>bit</i>	Rampa podignuta ako je RampOpen=1, RampOpen se postavlja kada vozač dobije karticu; RampOpen2 uzima u obzir stanje na PinEXTRampOpen pinu.
Event	<i>bit</i>	Setuje se kada novo vozilo dođe na rampu, resetuje kada se informacije (sati, minuti, sekunde, kategorija) pošalju Masteru. Pritom se rampa otvara.
BytesToReceive	<i>unsigned char</i>	Broj bajtova koji treba da se primi do dekodiranja, time se ostvaruje kontrola primljenih podataka
Command	<i>unsigned char</i>	bajt komande koji se prima
CommandModified	<i>unsigned char</i>	bajt koji se vraća Masteru
CallFlag, RTCSetupFlag, UpdateLCDFlag	<i>bit</i>	Oznaka da je stigao zahtev za prozivkom i Slejv treba da vrati odgovor; Oznaka da je stigao zahtev za podešavanjem vremena i Slejv treba da vrati odgovor; Oznaka da Slejv treba da inkrementira vreme za 1 sekundu.

## Raspored korišćenih funkcija

Naziv	Tip	Objašnjenje funkcija
init_variables()	<i>void</i>	inicijalizacija promenljivih
init()	<i>void</i>	podešava rad pinova mikrokontrolera, konfigurira periferije kontrolera, AD konvertor, prekide, kola tajmera; inicijalizuje UART
transmit()	<i>void</i>	obavlja slanje bajtova preko kola UART-a. Kao ulazni parameter funkcija uzima 8-bitni podatak.
UpdateLCD()	<i>void</i>	ispisuje stanje senzora, aktuatora i tajmera na LCD displej
processInputs()	<i>void</i>	prikuplja informacije sa senzora i menja stanja promenljivih koje čine status ulazne rampe. Funkcija se poziva periodično - nakon svake sekunde
interrupt()	<i>void</i>	koristi se za obradu prekida mikrokontrolera
main()	<i>void</i>	glavna funkcija programa

# Definisanje pinova i funkcija

```
// definicija ulaznih i izlaznih pinova

# define PinTaster PORTB.F0 // tasteri
# define PinSensor PORTB.F1

# define PinError PORTB.F4 // prekidaci:
# define PinEXTRampOpen PORTB.F5

# define PinOperation PORTA.F2 // dig.izlazi,
LED diode
# define PinRampOpen PORTA.F3
# define PinEvent PORTA.F4

# define DR PORTC.F5 // za upravljanje RS485
drajverom
```



```
void DecodeTime ();  
void ProcessInputs ();  
void IncrementTime ();  
void init ();  
void UpdateLCD ();
```

# Deklarisanje promenljivih

```
unsigned char RAMP_ID=0x00;
bit Operation;
bit Operation2;
bit RampOpen;
bit RampOpen2;
bit Event;
bit EXTRampOpen;
bit Error;
bit Sensor;
unsigned char Category=0x00;

// ID broj rampe
unsigned char BytesToReceive=0x00;
// broj bajtova koji treba da se primi do dekodiranja
unsigned char ch=0x00;
// primljeni bajt
unsigned char Command=0x00;
// bajt komande koji se prima
unsigned char CommandModified=0x00;
// bajt koji se salje
```

```
bit CallFlag;  
// oznaka da se stigao zahtev za prozivkom  
  
bit RTCSetupFlag;  
// oznaka da je stigao zahtev za podesavanjem  
vremena  
  
bit UpdateLCDFlag;  
  
bit TMP_Taster2;  
bit TMP_Taster1;  
  
bit TMP_Sensor2;  
bit TMP_Sensor1;  
  
bit TMP_Error2;  
bit TMP_Error1;  
  
bit TMP_EXTRampOpen2;  
bit TMP_EXTRampOpen1;
```

```
unsigned char Sec_X1=0x00;  
unsigned char Sec_X10=0x00;  
unsigned char Min_X1=0x00;  
unsigned char Min_X10=0x00;  
unsigned char Hour_X1=0x00;  
unsigned char Hour_X10=0x00;
```

```
unsigned char Tmp_Sec_X1=0x00;  
unsigned char Tmp_Sec_X10=0x00;  
unsigned char Tmp_Min_X1=0x00;  
unsigned char Tmp_Min_X10=0x00;  
unsigned char Tmp_Hour_X1=0x00;  
unsigned char Tmp_Hour_X10=0x00;
```

```
unsigned char Counter=0x00;  
unsigned char Counter2=0x00;
```

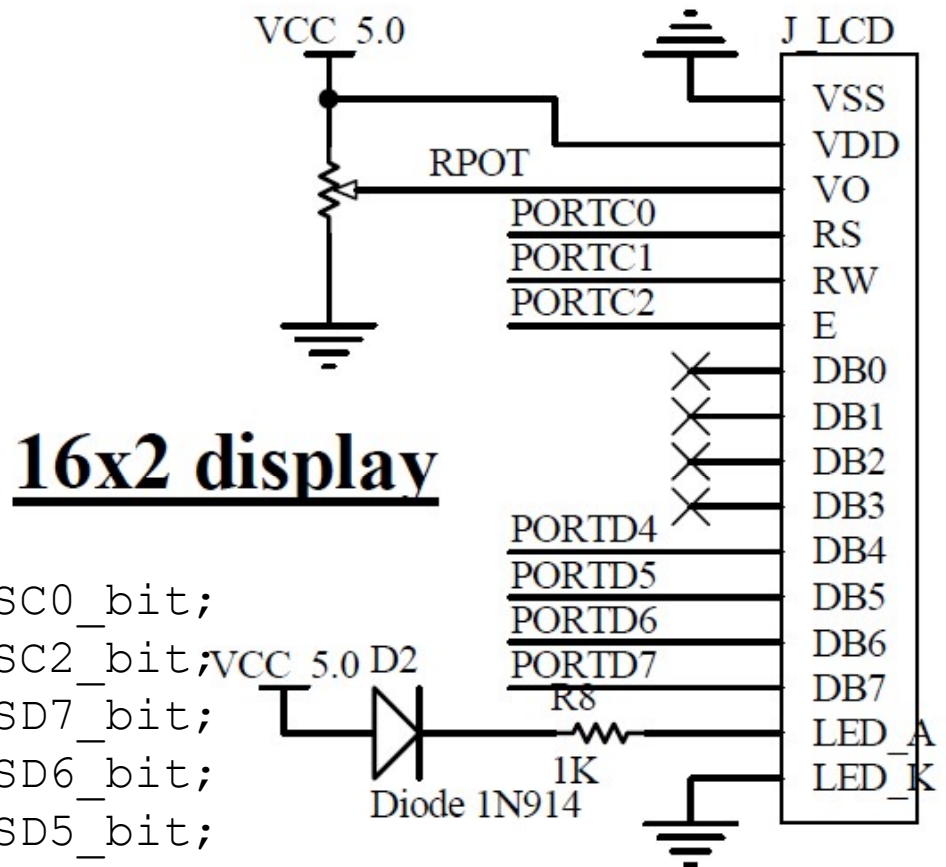
```
unsigned char Seconds=0x00;  
unsigned char Minutes=0x00;  
unsigned char Hours=0x00;
```

```
unsigned char X1;  
unsigned char X10;
```

# Deklaracija pinova koji se koriste za 16x2 LCD

```
// Lcd pinout settings
sbit LCD_RS at RC0_bit;
sbit LCD_EN at RC2_bit;
sbit LCD_D7 at RD7_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D4 at RD4_bit;

// Pin direction
sbit LCD_RS_Direction at TRISC0_bit;
sbit LCD_EN_Direction at TRISC2_bit;
sbit LCD_D7_Direction at TRISD7_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D4_Direction at TRISD4_bit;
```



```
void init_variables() {
```

```
    RAMP_ID=0x00;
```

```
    Category=0x00;
```

```
    Operation=0;
```

```
    Event=0;
```

```
    RampOpen=0;
```

```
    EXTRampOpen=0;
```

```
    Error=0;
```

```
    Sensor=0;
```

```
    Operation2=0;
```

```
    RampOpen2=0;
```

```
    BytesToReceive=0x00;
```

```
    ch=0x00;
```

```
    Command=0x00;
```

```
    CommandModified=0x00;
```

```
    Counter=0x00; Counter2=0x00;
```

```
    CallFlag=0; RTCSetupFlag=0;
```

```
    UpdateLCDFlag=0;
```

```
TMP_Taster2 = 0; TMP_Taster1 = 0;
```

```
TMP_Sensor2 = 0; TMP_Sensor1 = 0;
```

```
TMP_Error2 = 0; TMP_Error1 = 0;
```

```
TMP_EXTRampOpen2 = 0;
```

```
TMP_EXTRampOpen1 = 0;
```

```
    Sec_X1=0x00;
```

```
    Sec_X10=0x00;
```

```
    Min_X1=0x00;
```

```
    Min_X10=0x00;
```

```
    Hour_X1=0x00;
```

```
    Hour_X10=0x00;
```

```
    Tmp_Sec_X1=0x00;
```

```
    Tmp_Sec_X10=0x00;
```

```
    Tmp_Min_X1=0x00;
```

```
    Tmp_Min_X10=0x00;
```

```
    Tmp_Hour_X1=0x00;
```

```
    Tmp_Hour_X10=0x00;
```

```
    Seconds=0x00;
```

```
    Minutes=0x00;
```

```
    Hours=0x00;
```

```
    }  
20
```



```
void init() {  
  
    TRISA=0x03;  
    TRISB=0x3F;  
    TRISC=0xC0; // pinovi 6 i 7 su vezani za RS232  
    TRISD=0x0F;  
    TRISE=0x00;  
  
    PORTA=0x00;  
    PORTB=0x00;  
    PORTC=0x00;  
    PORTD=0x00;  
    PORTE=0x00;  
  
    ADCON0=0x00; // isključujemo A/D konverziju  
    ADCON1=0b00000110; // svi digitalni pinovi  
  
    INTCON = 0b11000000; // default  
    PIE1 = 0b00000000; // default  
}
```

```

T1CON=0b00110000;    // konfiguracija za Tajmer 1
// preskaler 1/8
TMR1H = 0x0B;    // startne vrednosti Tajmera 1
TMR1L = 0xDC;
T1CON.TMR1ON=1;

// Fosc=20MHz, Tosc= 50ns
// takt frekv. Fosc/4 dolazi na preskaler koji je
podesen na 1/8
// (10000)h- (0BDC)h= (F424)h= (62500)dec
// 62500 x 8 x 4 Tosc= 100ms

PIR1.TMR1IF = 0;
PIE1.TMR1IE = 1;
    Uart1_Init(19200);
TXSTA.TXEN=1;
RCSTA.SPEN=1;
RCSTA.CREN=1;
PIE1.RCIE=1;
INTCON.GIE=1;
} // od init()

```

```
void ProcessInputs () {

    TMP_Taster2 = TMP_Taster1;
    if (PinTaster == 1)
        TMP_Taster1 = 1; // aktivan 1
    else
        TMP_Taster1 = 0;
    if ((TMP_Taster2 == 1) && (TMP_Taster1 == 1))
    {
        Event = 1;
        PinEvent = 1;
    }

    TMP_Sensor2 = TMP_Sensor1;
    if (PinSensor == 1)
        TMP_Sensor1 = 1; // aktivan 1
    else
        TMP_Sensor1 = 0;
    if ((TMP_Sensor2 == 1) && (TMP_Sensor1 == 1))
        RampOpen = 0;
}
```

```
TMP_Error2 = TMP_Error1;
if (PinError == 1)
    TMP_Error1 = 1; // aktivan 1
else
    TMP_Error1 = 0;
if ((TMP_Error2 == 1) && (TMP_Error1 == 1))
    Error = 1;
else if ((TMP_Error2 == 0) && (TMP_Error1 == 0))
    Error = 0;

TMP_EXTRampOpen2 = TMP_EXTRampOpen1;
if (PinEXTRampOpen == 1)
    TMP_EXTRampOpen1 = 1; // aktivan 1
else
    TMP_EXTRampOpen1 = 0;

if ((TMP_EXTRampOpen2 == 1) && (TMP_EXTRampOpen1 == 1))
    EXTRampOpen = 1;
else if ((TMP_EXTRampOpen2 == 0) && (TMP_EXTRampOpen1 == 0))
    EXTRampOpen = 0;
```

```
if (PORTD.F2 == 1)
    if (PORTD.F3 == 1)
        Category = 0x03;
    else
        Category = 0x02;
else if (PORTD.F3 == 1)
    Category = 0x01;
else
    Category = 0x00;
}
```

```

void main() {
    init();
    init_variables();
    Lcd_Init();
    Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off
    UpdateLCD();

    while (1) {
        RampOpen2=RampOpen|EXTRampOpen;
        if (Error==1) Operation2=0;
        else Operation2=Operation;
        PinOperation=Operation2;
        PinRampOpen=RampOpen2;

        if (UpdateLCDFlag==1) {
            UpdateLCDFlag=0;
            UpdateLCD();
        }

        if ((BytesToReceive>0) && (Counter2==0)) {
            BytesToReceive=0;
        }
    }
}

```



# Master

## Prozivka

KOMANDA		ID	RAMPE	$Y_{\epsilon}\{0,1\}$			
0	0	1	X	Y	Y	Y	Y

Podešavanje RTC. Posle ovog bajta, Slejv automat šalje još tri dodatna bajta: *sekunde, minuti, sati*

KOMANDA		ID	RAMPE	$Y_{\epsilon}\{0,1\}$			
0	1	1	X	Y	Y	Y	Y

# Slejv

Automat nema kartica.

KOMANDA		ID	RAMPE	$Y_{\epsilon}\{0,1\}$			
0	0	0	X	Y	Y	Y	Y

Automat radi ali nije pritisnut taster

KOMANDA		ID	RAMPE	$Y_{\epsilon}\{0,1\}$			
0	0	1	X	Y	Y	Y	Y

Pritisnut je taster, tj. prošlo je vozilo kroz rampu. Posle ovog bajta, Slejv automat šalje još četiri dodatna bajta: *sekunde, minuti, sati i kategorija vozila.*

KOMANDA		ID	RAMPE	$Y_{\epsilon}\{0,1\}$			
0	1	0	X	Y	Y	Y	Y

Odgovor da je RTC podešen

KOMANDA		ID	RAMPE	$Y_{\epsilon}\{0,1\}$			
0	1	1	X	Y	Y	Y	Y

```

if (CallFlag==1) {
    if ((Command&0x10)==0x10) Operation=1;
    else Operation=0;

    if (Error==1) {    // nema kartica
        DR=1;
        if (Operation==1)
            CommandModified=0b00010000|RAMP_ID;
        else CommandModified=0b00000000|RAMP_ID;
        transmit(CommandModified);
        DR=0;
    }
    else if (Event==0) {
// taster nije pritisnut, vraca se prozivka
        DR=1;
        if (Operation==1)
            CommandModified=0b00110000|RAMP_ID;
        else CommandModified=0b00100000|RAMP_ID;
        transmit(CommandModified);
        DR=0;
    }
}

```

```

    else {
// taster je pritisnut
// vraca se bajt koji znaci da je taster pritisnut
// vraca se inf. o vremenu - sekundi, minuti i sati
// i kategorija vozila

        if (Operation==0x01) {
            DR=1;
            CommandModified=0b01010000|RAMP_ID;
            transmit(CommandModified);
            transmit(Seconds);
            transmit(Minutes);
            transmit(Hours);
            transmit(Category);
            DR=0;
            Event=0; PinEvent=0;
            RampOpen=1;
        }
    }
    CallFlag=0;
} // od if (CallFlag==1){

```

```

    if (RTCSetupFlag==1) {

        Sec_X1= Tmp_Sec_X1;
        Sec_X10= Tmp_Sec_X10;
        Min_X1= Tmp_Min_X1;
        Min_X10= Tmp_Min_X10;
        Hour_X1= Tmp_Hour_X1;
        Hour_X10= Tmp_Hour_X10;

        DR=1;

        if (Operation==1)
CommandModified=0b01110000|RAMP_ID;
        else  CommandModified=0b01100000|RAMP_ID;

        transmit(CommandModified);
        RTCSetupFlag=0;
        DR=0;
    }
} // od while (1) {
} // od main()

```