Računske vežbe iz Projektovanja Elektronskih Sistema
cas 8

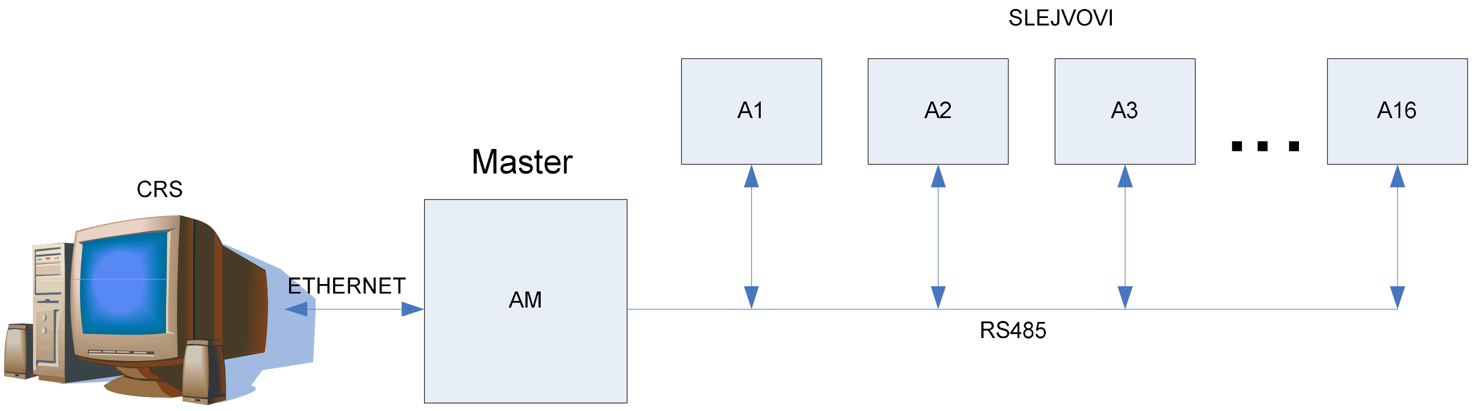
Doc.dr Borisav Jovanović

Sadržaj:

- **Realizacija firmvera Slejv automata 2.**
- Opis *interrupt()* funkcije, opis komunikacionih funkcija za rad sa UART-om,
- opis ostalih funkcija.

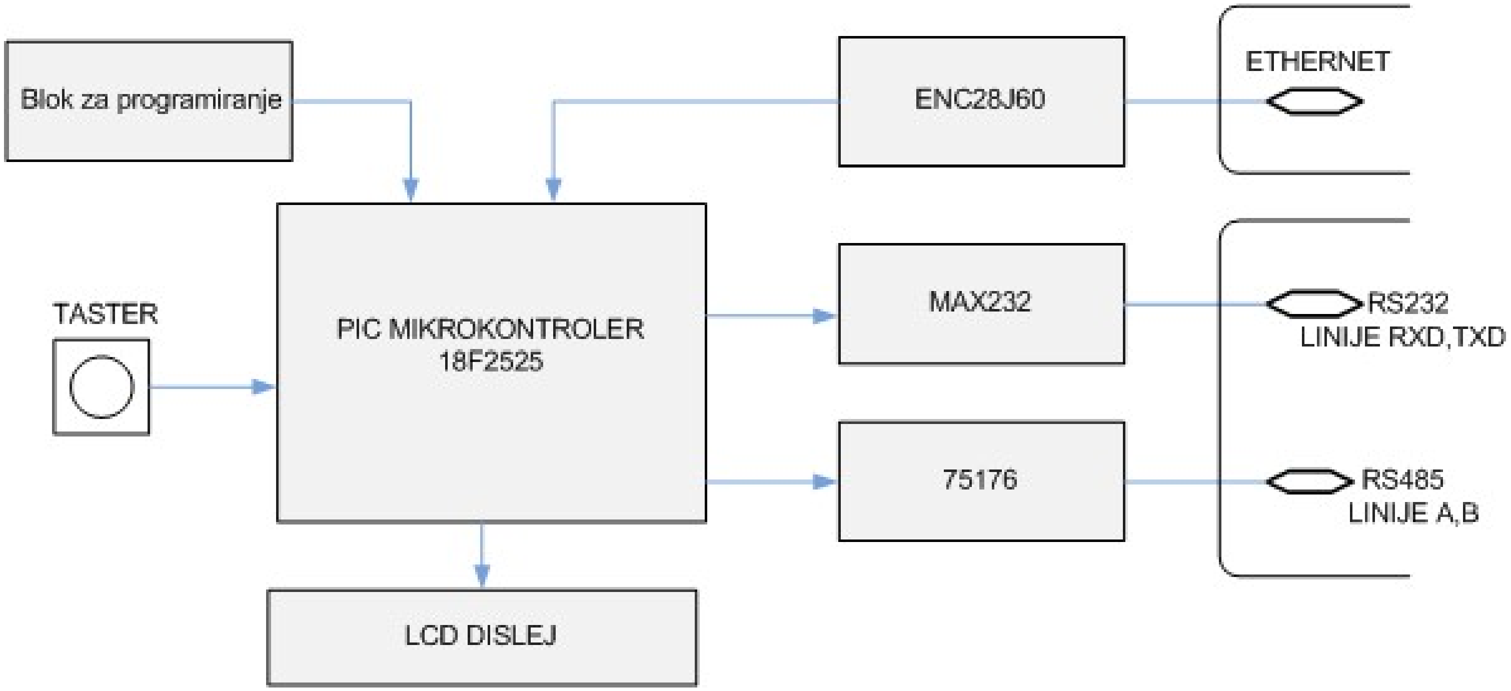
Realizacija Master automata u C-u za mikrokontroler PIC18F2525

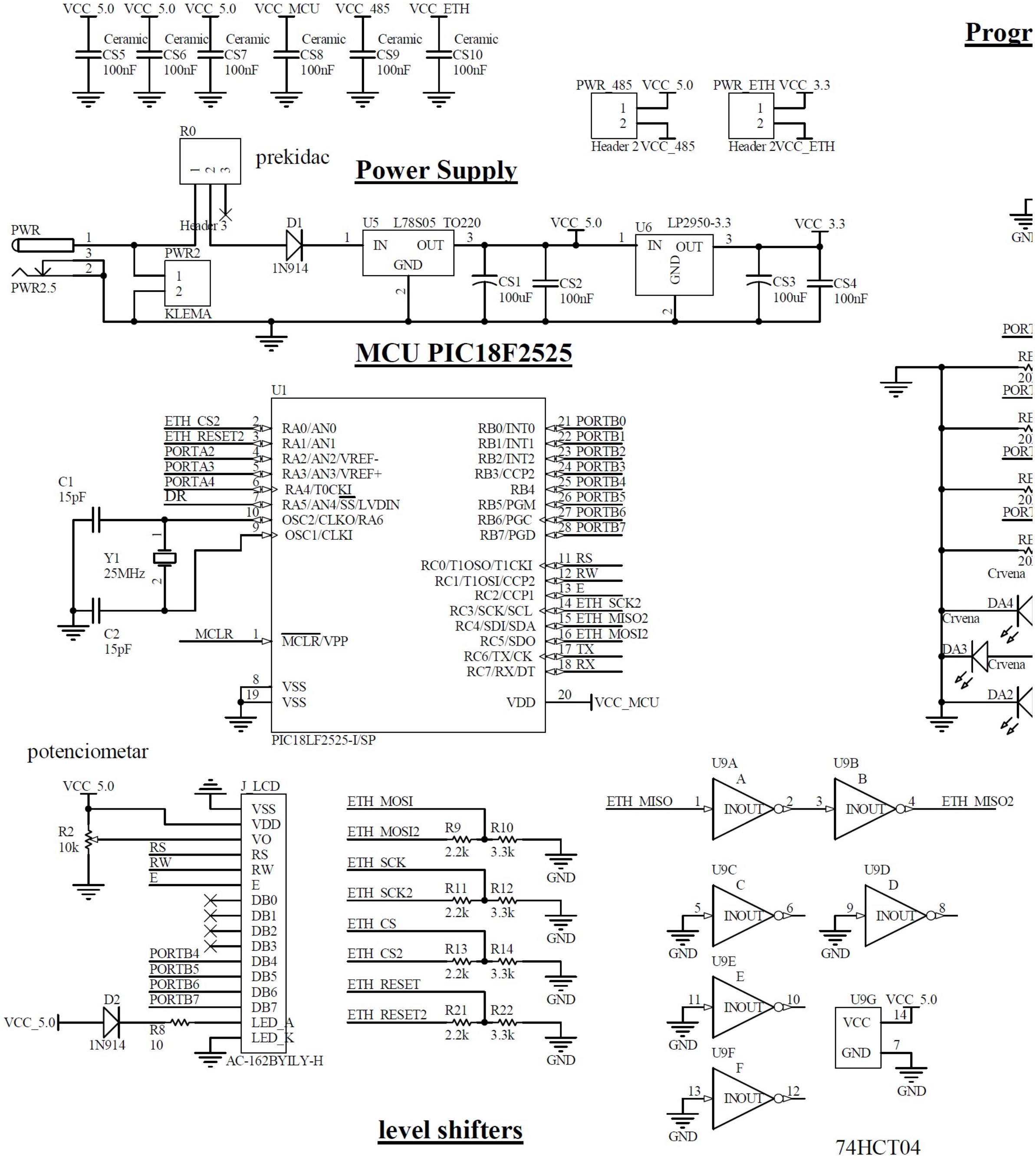
Arhitektura Sistema

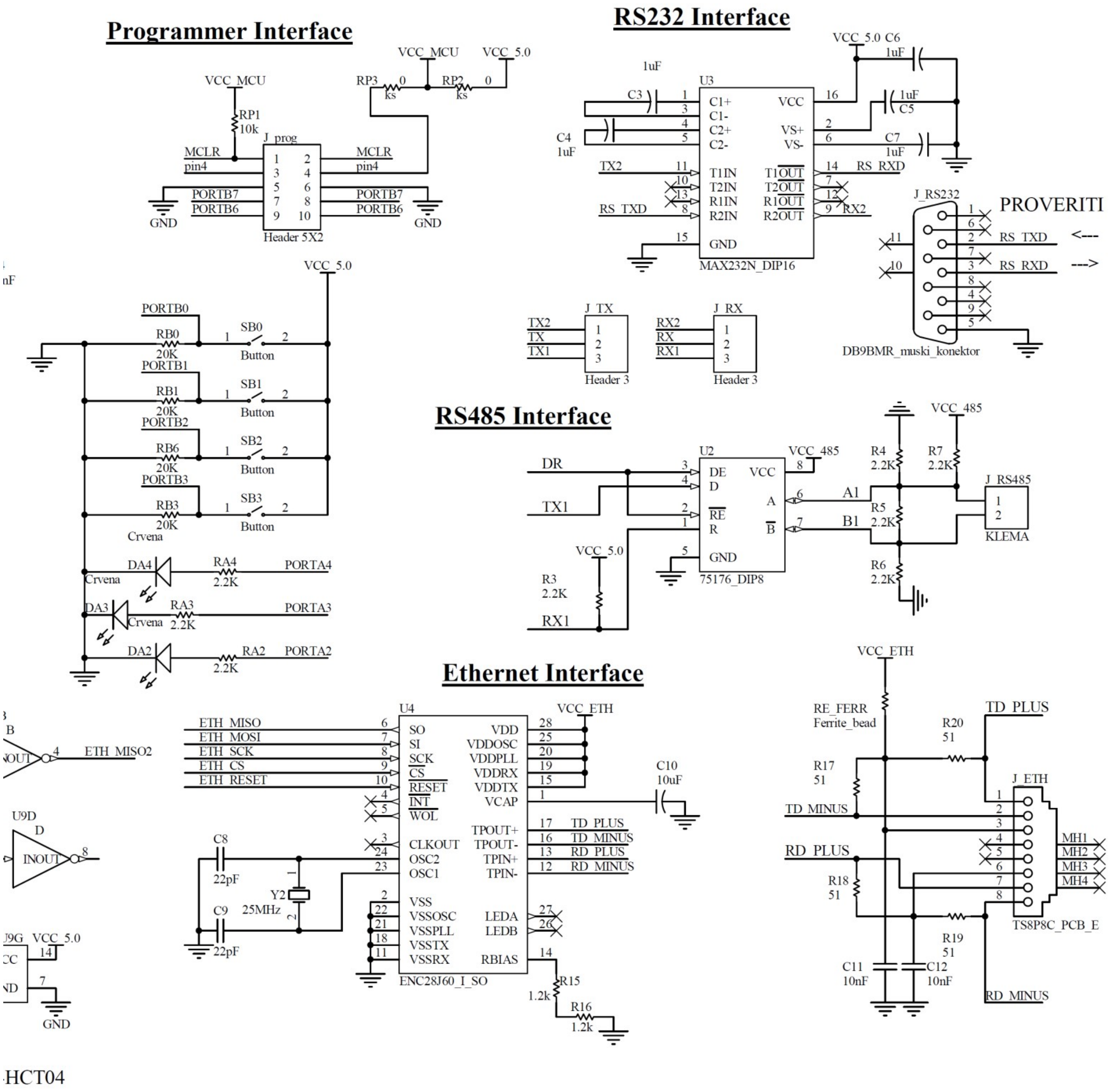


Osnovne komponente sistema su:

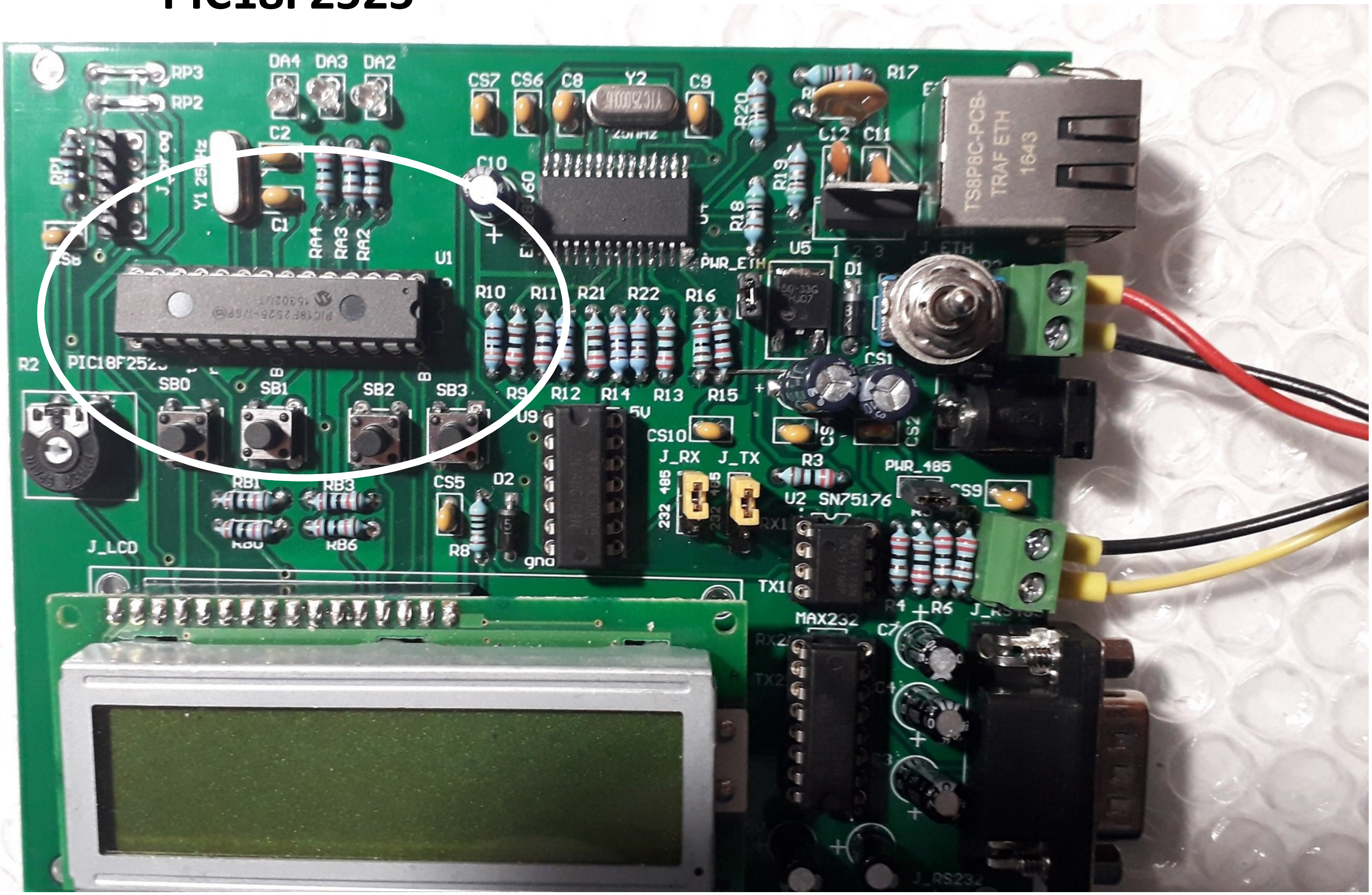
- CRS (Centralni Računarski Sistem)
- Master automat (AM)
- Slejv automati, koje se vezuju za svako ulazno mesto na autoputu (A1-A16).

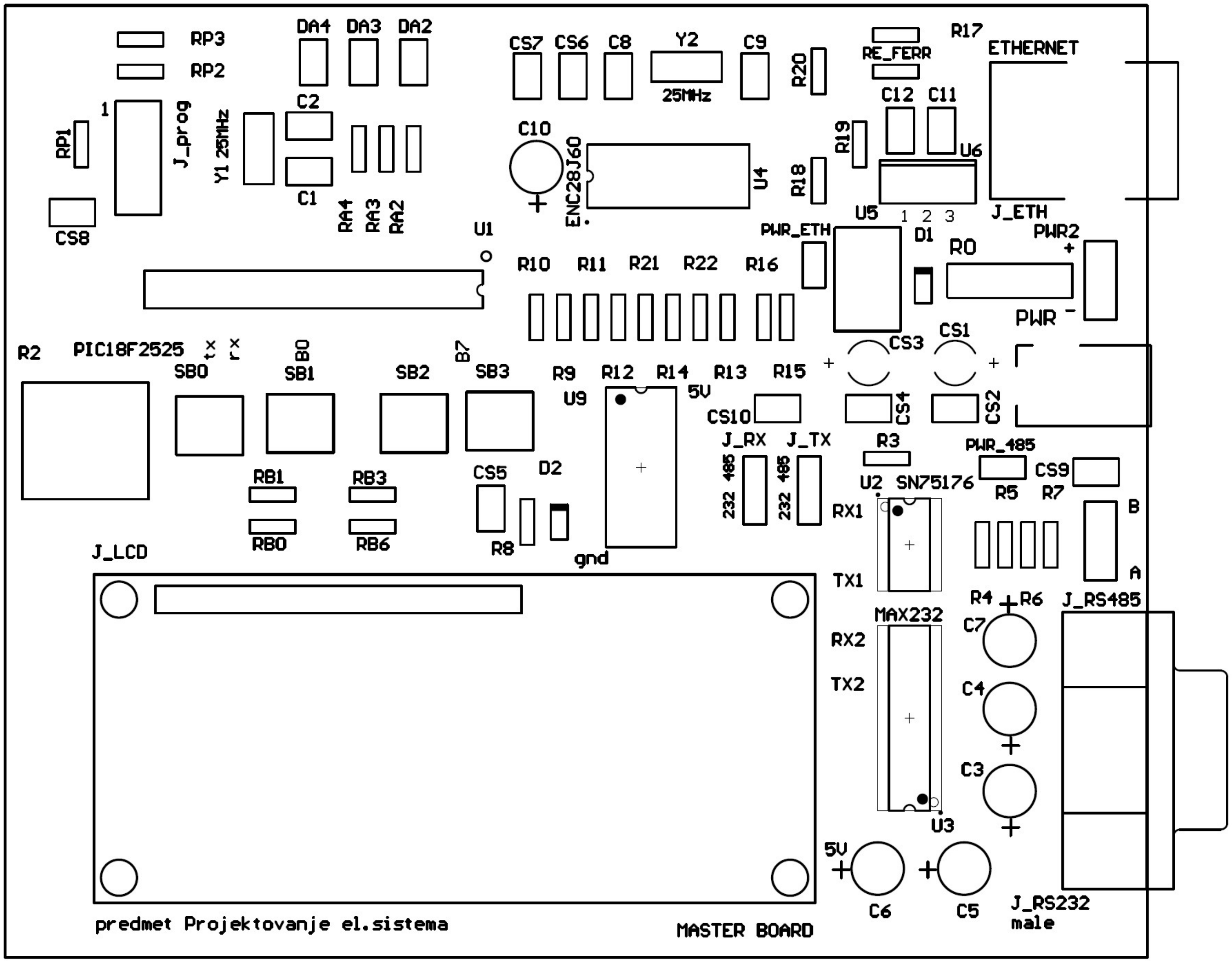

Blok za programiranje
ENC28J60
ETHERNET
TASTER
PIC MIKROKONTROLER
18F2525
MAX232
RS232
LINIJE RXD,TXD
75176
RS485
LINIJE A,B
LCD DISLEJ


Progr
Power Supply
MCU PIC18F2525
level shifters
VCC_5.0 VCC_5.0 VCC_5.0 VCC_MCU VCC_485 VCC_ETH
Ceramic CS5 100nF
Ceramic CS6 100nF
Ceramic CS7 100nF
Ceramic CS8 100nF
Ceramic CS9 100nF
Ceramic CS10 100nF
PWR_485 VCC_5.0
Header 2 VCC_485
PWR_ETH VCC_3.3
Header 2VCC_ETH
R0
prekidac
Header 3
PWR
PWR2.5
PWR2
KLEMA
D1
1N914
U5 L78S05_TO220
IN OUT GND
VCC_5.0
U6 LP2950-3.3
IN OUT GND
VCC_3.3
CS1 100uF
CS2 100nF
CS3 100uF
CS4 100nF
GND
U1
ETH_CS2
ETH_RESET2
PORTA2
PORTA3
PORTA4
DR
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS/LVDIN
OSC2/CLKO/RA6
OSC1/CLKI
C1 15pF
Y1 25MHz
C2 15pF
MCLR
MCLR/VPP
VSS
VSS
RB0/INT0 21 PORTB0
RB1/INT1 22 PORTB1
RB2/INT2 23 PORTB2
RB3/CCP2 24 PORTB3
RB4 25 PORTB4
RB5/PGM 26 PORTB5
RB6/PGC 27 PORTB6
RB7/PGD 28 PORTB7
RC0/T1OSO/T1CKI 11 RS
RC1/T1OSI/CCP2 12 RW
RC2/CCP1 13 E
RC3/SCK/SCL 14 ETH_SCK2
RC4/SDI/SDA 15 ETH_MISO2
RC5/SDO 16 ETH_MOSI2
RC6/TX/CK 17 TX
RC7/RX/DT 18 RX
VDD 20 VCC_MCU
PIC18LF2525-I/SP
PORT
RE
20
Crvena
DA4
DA3
DA2
potenciometar
VCC_5.0
R2 10k
J_LCD
VSS VDD VO RS RW E DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7 LED_A LED_K
RS RW E
PORTB4 PORTB5 PORTB6 PORTB7
D2 1N914
R8 10
AC-162BYILY-H
ETH_MOSI
ETH_MOSI2 R9 2.2k R10 3.3k
ETH_SCK
ETH_SCK2 R11 2.2k R12 3.3k
ETH_CS
ETH_CS2 R13 2.2k R14 3.3k
ETH_RESET
ETH_RESET2 R21 2.2k R22 3.3k
U9A A
U9B B
ETH_MISO
ETH_MISO2
INOUT
U9C C
U9D D
U9E E
U9F F
U9G VCC_5.0
VCC GND
74HCT04


Programmer Interface
VCC_MCU
RP1
10k
J_prog
MCLR
pin4
PORTB7
PORTB6
GND
Header 5X2
VCC_MCU
VCC_5.0
RP3 0 ks
RP2 0 ks
RS232 Interface
VCC_5.0 C6
1uF
U3
C3
C4
1uF
C1+
C1-
C2+
C2-
VCC
VS+
VS-
C5
C7
TX2
T1IN
T2IN
R1IN
R2IN
T1OUT
T2OUT
R1OUT
R2OUT
RS_RXD
RS_TXD
RX2
GND
MAX232N_DIP16
J_RS232
PROVERITI
RS_TXD <---
RS_RXD --->
DB9BMR_muski_konektor
J_TX
TX2
TX
TX1
Header 3
J_RX
RX2
RX
RX1
Header 3
VCC_5.0
PORTB0
RB0 20K
SB0 Button
PORTB1
RB1 20K
SB1 Button
PORTB2
RB6 20K
SB2 Button
PORTB3
RB3 20K
SB3 Button
Crvena
DA4
RA4 2.2K
PORTA4
Crvena
DA3
RA3 2.2K
Crvena
PORTA3
DA2
RA2 2.2K
PORTA2
RS485 Interface
DR
TX1
RX1
VCC_5.0
R3 2.2K
U2
DE
D
RE
R
GND
VCC
A
B
VCC_485
75176_DIP8
A1
B1
R4 2.2K
R7 2.2K
R5 2.2K
R6 2.2K
VCC_485
J_RS485
KLEMA
Ethernet Interface
U4
ETH_MISO
ETH_MOSI
ETH_SCK
ETH_CS
ETH_RESET
SO
SI
SCK
CS
RESET
INT
WOL
CLKOUT
OSC2
OSC1
VSS
VSSOSC
VSSPLL
VSSTX
VSSRX
VDD
VDDOSC
VDDPLL
VDDRX
VDDTX
VCAP
TPOUT+
TPOUT-
TPIN+
TPIN-
LEDA
LEDB
RBIAS
VCC_ETH
C10 10uF
TD_PLUS
TD_MINUS
RD_PLUS
RD_MINUS
C8 22pF
C9 22pF
Y2 25MHz
ENC28J60_I_SO
R15 1.2k
R16 1.2k
VCC_ETH
RE_FERR
Ferrite_bead
R17 51
R18 51
R19 51
R20 51
TD_PLUS
TD_MINUS
RD_PLUS
RD_MINUS
J_ETH
MH1
MH2
MH3
MH4
TS8P8C_PCB_E
C11 10nF
C12 10nF
B
INOUT
ETH_MISO2
U9D
D
INOUT
U9G VCC_5.0
CC
ND
GND
HCT04

PIC18F2525


RP3
RP2
RP1
DA4
DA3
DA2
CS7
CS6
C8
Y2
C9
R20
R17
C12
C11
R19
R18
U5
D1
C2
C1
Y1 25MHz
RA4
RA3
RA2
C10
U1
J_ETH
PWR_ETH
TS8P8C-PCB-TRAF ETH 1643
R10
R11
R21
R22
R16
CS1
CS2
R2
PIC18F2525
SB0
SB1
SB2
SB3
R9
R12
R14
R13
R15
U9
5V
CS10
J_RX
J_TX
R3
PWR_485
CS9
RB1
RB3
RB0
RB6
CS5
D2
R8
gnd
232
485
U2
SN75176
RX1
TX1
R4
R6
J_LCD
MAX232
RX2
TX2
C7
C4
C3
J_RS232

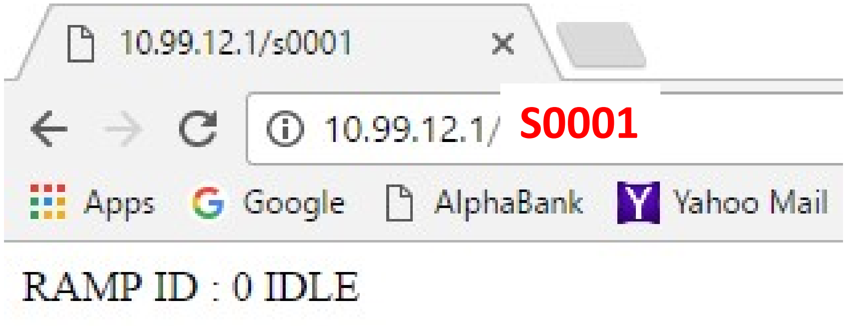

RP3
RP2
1
RP1
J_prog
CS8
Y1 25MHz
DA4
DA3
DA2
C2
C1
RA4
RA3
RA2
CS7
CS6
C8
Y2
25MHz
C9
C10
ENC28J60
U4
R20
R19
R18
R17
RE_FERR
C12
C11
U6
ETHERNET
J_ETH
U5
1 2 3
U1
R10
R11
R21
R22
R16
PWR_ETH
D1
R0
PWR2
+
-
PWR
R2
PIC18F2525
tx
rx
B0
B7
SB0
SB1
SB2
SB3
R9
R12
R14
R13
R15
U9
5V
CS3
CS1
+
+
CS10
CS4
CS2
J_RX
J_TX
R3
PWR_485
232 485
232 485
U2
SN75176
CS9
R5
R7
B
A
RX1
TX1
RB1
RB3
RB0
RB6
CS5
D2
R8
gnd
J_LCD
R4
R6
C7
J_RS485
MAX232
RX2
TX2
C4
C3
U3
5V
C6
C5
J_RS232
male
predmet Projektovanje el.sistema
MASTER BOARD

a)

KOMANDA	Slejv:15-12		Slejv: 11- 8		Slejv: 7- 4		Slejv: 3-0	
‘s’	3	X	3	X	3	X	3	X

b)

KOMANDA	SEKUNDE	MINUTI	SATI
‘r’			



RAMP ID : 0 IDLE

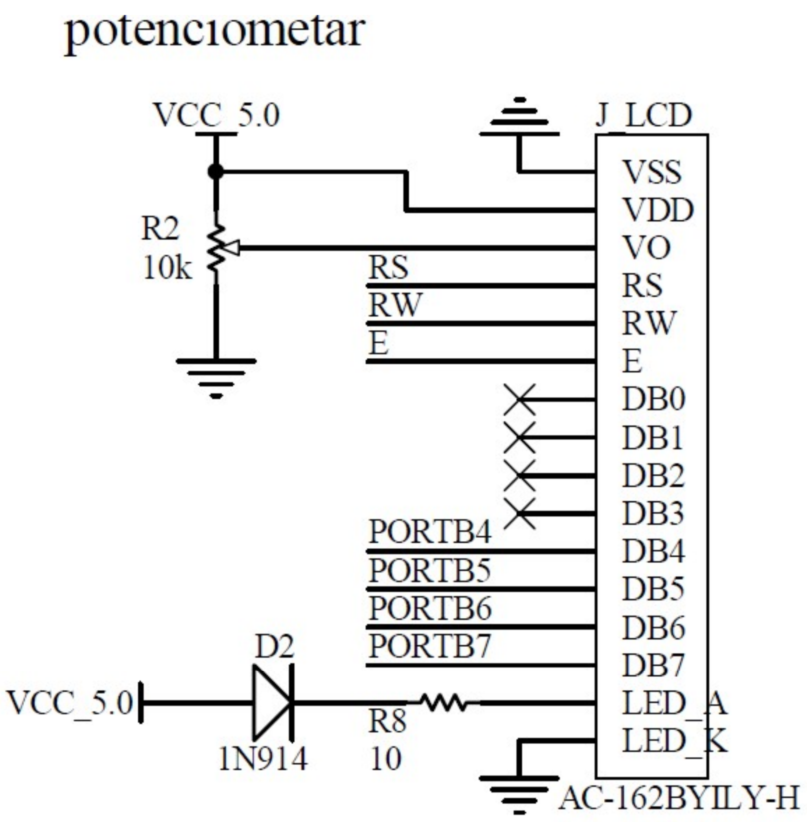
primer “s0001” – samo Slejv sa ID brojem 0 je operativan

xɛ{0,1}, kontrola slejva				15	14	13	12	xɛ{0,1}, kontrola slejva				11	10	9	8	xɛ{0,1}, kontrola slejva				7	6	5	4	xɛ{0,1}, kontrola slejva				3	2	1	0
0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	1

```
#define SPI_Ethernet_HALFDUPLEX     0
#define SPI_Ethernet_FULLDUPLEX     1
#define DR  PORTA.F5

typedef struct {
  unsigned canCloseTCP: 1;  // flag which closes socket
  unsigned isBroadcast: 1;
  // flag which denotes that the IP package has been received via subnet broadcast address
} TEthPktFlags;

const unsigned char httpHeader[] = "HTTP/1.1 200 OK\nContent-type: " ;
 // HTTP header
const unsigned char httpMimeTypeHTML[] = "text/html\n\n" ;
 // HTML MIME type
const unsigned char httpMimeTypeScript[] = "text/plain\n\n" ;
// TEXT MIME type
unsigned char httpMethod[] = "GET /";

// mE ethernet NIC pinout
sfr sbit SPI_Ethernet_Rst at RA1_bit;
sfr sbit SPI_Ethernet_CS  at RA0_bit;
sfr sbit SPI_Ethernet_Rst_Direction at TRISA1_bit;
sfr sbit SPI_Ethernet_CS_Direction  at TRISA0_bit; // end Ethernet NIC definitions
```

```
unsigned char   myMacAddr[6] = {0x00, 0x14, 0xA5, 0x76, 0x19, 0x3f} ;
       // MAC adresa uredaja
unsigned char   myIpAddr[4] = {10, 99, 12, 1} ;
       //IP adresa uredaja
unsigned char   getRequest[15] ;    //  HTTP request buffer
unsigned char   dyna[31] ;          //  buffer for dynamic response
unsigned long   httpCounter = 0 ;   //  counter of HTTP requests

unsigned char  i, brojac, RAMP_ID, Flag1, Flag2, Flag3, ch, OBB;
unsigned char  niz[150];
unsigned char  br_ch;
unsigned char  seconds, minutes, hours;

 // nizovi za pojedinacne rampe
 unsigned char Operation[16];
 unsigned char Comm[16];
 unsigned char Cmd[16];
 unsigned char Cat[16];
 unsigned char Hour[16];
 unsigned char Min[16];
 unsigned char Sec[16];
```

```
sbit LCD_RS at RC0_bit; // Lcd pinout settings
sbit LCD_RW at RC1_bit;
sbit LCD_EN at RC2_bit;
sbit LCD_D7 at RB7_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D4 at RB4_bit;

sbit LCD_RS_Direction at TRISC0_bit; // Pin direction
sbit LCD_RW_Direction at TRISC1_bit;
sbit LCD_EN_Direction at TRISC2_bit;
sbit LCD_D7_Direction at TRISB7_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB4_bit;
```



```
unsigned char * p_ch=0x00;
// Pokazivac na prvi karakter koji koristimo prilikom ispisivanja stringova

unsigned char pom_ch, pom_des, pom_jed;
unsigned char pom_nula=0x30;
```

```
void init_variables(){
  br_ch=0x00;
  OBB=0x00;
  Flag1=0x00;
  Flag2=0x00;
  Flag3=0x00;
  brojac=0x00;
  RAMP_ID = 0x0F;
  for (i=0;i<150;i++) niz[i]=0x00;
  for (i=0;i<16;i++){
    Operation[i]=0x00;
    Comm[i]=0x00;
    Cmd[i]=0x00;
    Cat[i]=0x00;
    Hour[i]=0x00;
    Min[i]=0x00;
    Sec[i]=0x00;
  }
}
```

```
void init ()
{
    PIR1 = 0b00000000; // dozvola prijema i predaje preko EUSART-a
    PIE1 = 0b00100001; // dozvola prekida za EUSART, RCIE, TMR1IE
    //PIE1.TMR1IE = 1;
    //PIE1.RC1IE=1;

    T1CON=0b10110000;   // konfiguracija za tajmer1
    T1CON.TMR1ON=1;
    // 16-bit operation
    // preskaler 1:8
    // 25MHz T0=40ns
    // 40ns*4*8=1.28us
    // 25ms=25000us=1.28*19531=  B#B%
    TMR1L = 0xB5;
    TMR1H = 0xB3;

    INTCON = 0b01000000;  // periferisjki interapt
    INTCON.GIE=1; // globalna dozvola prekida
    TRISA=0x00;
    TRISB=0x0F;
    TRISC=0xD0; // 0b11010000;
    PORTA=0x00;
    PORTB=0x00;
    PORTC=0x00;

    ADCON0=0x00; // iskljucujemo A/D konverziju
    ADCON1=0x0F; // svi digitalni
    UART1_Init(19200); // konfigurišemo brzinu od 19200
    TXSTA.TXEN=1;
    RCSTA.SPEN=1;
    RCSTA.CREN=1;

    Lcd_Init();
    Lcd_Cmd(_LCD_CURSOR_OFF);
    postaviPortove();

    SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV64, _SPI_DATA_SAMPLE_MIDDLE,
_SPI_CLK_IDLE_LOW, _SPI_LOW_2_HIGH);

    SPI_Ethernet_Init(myMacAddr, myIpAddr, SPI_Ethernet_FULLDUPLEX);

}
```

```
void UpdateLCD()
{
  int i = 0;
  Lcd_Out(1, 1, "Operation     ");
  for (i = 0; i <= 15; i++)
  {
    if (Operation[i] == 1)
      Lcd_Chr(2, 16 - i, '1');
    else
      Lcd_Chr(2, 16 - i, '0');
  }
}
```

```
void main(void) {
  unsigned char ByteX = 0x00;
  init();
  init_variables();
 while (1)
   {
       SPI_Ethernet_doPacket() ;
       if (Flag1==0x01) { //ovde se ulazi na svakih 125ms
           Flag1=0x00; //vratimo FLAG1 na nulu
           RAMP_ID++;               // biramo sledeci slejv
           if (RAMP_ID == 0x10){
           //kada se prozovu svih 16 rampi onda se brojac rampi vrati na nulu
                RAMP_ID = 0x00;
                PORTA.F4=1; //pali se dioda na svake 2 sekunde,
                 //16x125ms=2s
                 if (Flag3==0x01) {
                    Flag3=0x00;
                    Flag2=0x00;
                  }
                  //FLAG 3 se postavlja na 0 ako se proziva rampa a na 1
                 // ako se podešava sat realnog vremena
                 else if (Flag2==0x01) Flag3=0x01;
                 UpdateLCD();
           }
           else PORTA.F4=0;
if (Flag3 == 0x00) {  //   salje se prozivka rampama
      DR = 1;
      if (Operation[RAMP_ID] == 0x01)  ByteX = 0x30 + RAMP_ID;
      else ByteX = 0x20 + RAMP_ID;
      transmit(ByteX);
      DR = 0;
      OBB = 0x05; //OBB (ocekivani broj bajtova) postavlja se na 5,
      //sto znaci da sledeci bajt koji primamo predstavlja komandu
 } // od if (Flag3==0x00
  else { //ovde se salje svim rampama zahtev za podesavanje vremena,
      DR = 1;
      ByteX = 0x70 + RAMP_ID;
      transmit(ByteX); // komandni
      transmit(seconds);
      transmit(minutes);
      transmit(hours);
      DR = 0;
      OBB = 0x05; // opet se OBB postavlja na 5
     }          // od else
   }            // od Flag1==1
  }             // od  while (1)
} // od main()
```

```
void interrupt () {  // korišceni su prekid serijske komunikacije i tajmera 1

   if  ((PIE1.TMR1IE==1) && (PIR1.TMR1IF==1)){
    // prekid Tajmera 1 na svakih 25ms
    PIE1.TMR1IE = 1;
    PIR1.TMR1IF = 0;
    if (brojac == 0x04) {  // na svakih 125ms proziva se po jedna rampa
      brojac = 0x00;
      Flag1=0x01; // podiže se flag koji nam govori da je
       // došlo vreme da se prozove rampa,
    }
    else {
        brojac++;
    }
    TMR1L = 0xB5;
    TMR1H = 0xB3;
   }
```

Master

Prozivka

KOMANDA				ID RAMPE Yϵ{0,1)			
0	0	1	X	Y	Y	Y	Y

Slejv

Automat nema kartica.

KOMANDA				ID RAMPE Yϵ{0,1)			
0	0	0	X	Y	Y	Y	Y

Automat radi ali nije pritisnut taster

KOMANDA				ID RAMPE Yϵ{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 Yϵ{0,1)			
0	1	0	X	Y	Y	Y	Y

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

KOMANDA				ID RAMPE Yϵ{0,1)			
0	1	1	X	Y	Y	Y	Y

Odgovor da je RTC podešen

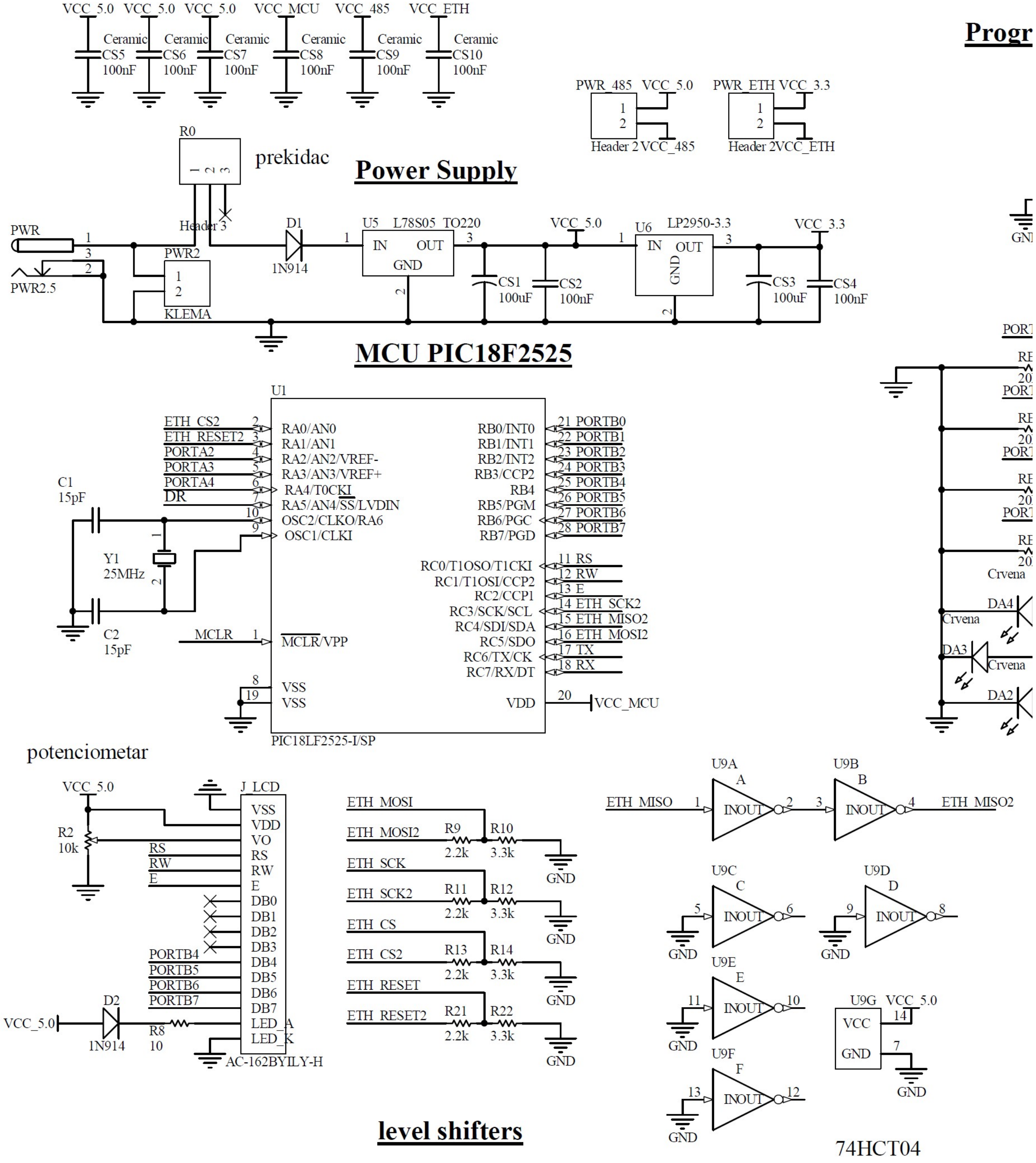
KOMANDA				ID RAMPE Yϵ{0,1)			
0	1	1	X	Y	Y	Y	Y

```
if ((PIE1.RCIE) && (PIR1.RCIF)){
// prekid  serijske komunikacije

    unsigned char ch;
    PIR1.RCIF = 0;
    ch=RCREG;  // prima se bajt preko UART-a
    if (OBB!=0x00) {
      if (OBB==0x05) {
// prijem bajta komande, koja se dekodira,
// i onda se odreduje da li treba još da se primaju bajtovi
         Comm[RAMP_ID]=1; // komunikacija je OK
         if ((ch & 0xE0)== 0x00) {OBB=0x00; Cmd[RAMP_ID]=3;}  // NO CARDS
         if ((ch & 0xE0)== 0x20) OBB=0x00;  // IDLE
         if ((ch & 0xE0)== 0x40) {OBB=0x04; Cmd[RAMP_ID]=1;}  // VEHICLE
         if ((ch & 0xE0)== 0x60) {OBB=0x00; Cmd[RAMP_ID]=2;}  // RTC
      }
```

```
 else {
         switch (OBB) {
          //   sekunde, minuti, sati i kategorija vozila koje je prošlo
              case 4: Sec[RAMP_ID]=ch;  break;  //ch_sec=ch;
              case 3: Min[RAMP_ID]=ch;  break;  //ch_min=ch;
              case 2: Hour[RAMP_ID]=ch; break; //ch_hour=ch;
              case 1: Cat[RAMP_ID]=ch; break; //ch_cat=ch;
              default: break;
         }
        OBB--;
      }
    }
   }  //  if ((PIE1.RCIE) && (PIR1.RCIF)){
} //void interrupt ()
```

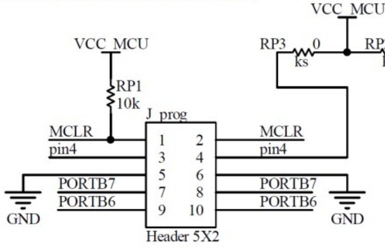
Sadržaj:

- **Realizacija firmvera Master automata deo 2.**, Opis *interrupt()* funkcije,
- opis komunikacionih funkcija za rad sa *Ethernet*-om i UART-om,
- opis ostalih funkcija.

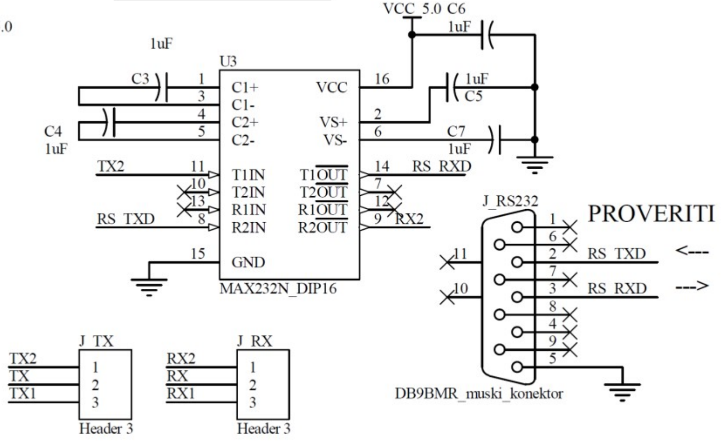


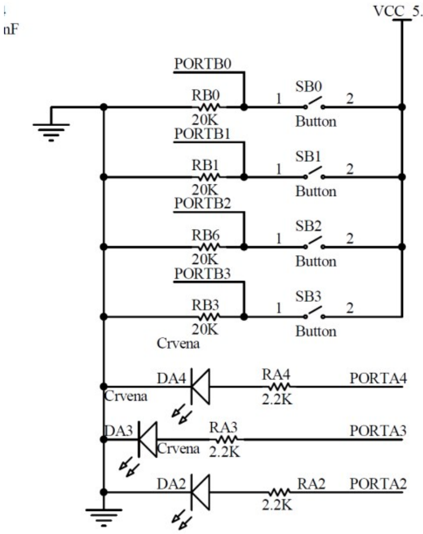
VCC_5.0
VCC_5.0
VCC_5.0
VCC_MCU
VCC_485
VCC_ETH
Ceramic
CS5
100nF
Ceramic
CS6
100nF
Ceramic
CS7
100nF
Ceramic
CS8
100nF
Ceramic
CS9
100nF
Ceramic
CS10
100nF
Progr
PWR_485
VCC_5.0
1
2
Header 2
VCC_485
PWR_ETH
VCC_3.3
1
2
Header 2
VCC_ETH
R0
1 2 3
prekidac
Header 3
Power Supply
PWR
PWR2.5
PWR2
KLEMA
D1
1N914
U5
L78S05_TO220
IN
OUT
GND
VCC_5.0
U6
LP2950-3.3
IN
OUT
GND
VCC_3.3
CS1
100uF
CS2
100nF
CS3
100uF
CS4
100nF
GND
MCU PIC18F2525
U1
ETH_CS2
ETH_RESET2
PORTA2
PORTA3
PORTA4
DR
RA0/AN0
RA1/AN1
RA2/AN2/VREF-
RA3/AN3/VREF+
RA4/T0CKI
RA5/AN4/SS/LVDIN
OSC2/CLKO/RA6
OSC1/CLKI
C1
15pF
Y1
25MHz
C2
15pF
MCLR
MCLR/VPP
VSS
VSS
RB0/INT0
RB1/INT1
RB2/INT2
RB3/CCP2
RB4
RB5/PGM
RB6/PGC
RB7/PGD
21 PORTB0
22 PORTB1
23 PORTB2
24 PORTB3
25 PORTB4
26 PORTB5
27 PORTB6
28 PORTB7
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
11 RS
12 RW
13 E
14 ETH_SCK2
15 ETH_MISO2
16 ETH_MOSI2
17 TX
18 RX
VDD
20
VCC_MCU
PIC18LF2525-I/SP
PORT
RE
20
Crvena
DA4
Crvena
DA3
Crvena
DA2
potenciometar
VCC_5.0
R2
10k
J_LCD
VSS
VDD
VO
RS
RW
E
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
LED_A
LED_K
RS
RW
E
PORTB4
PORTB5
PORTB6
PORTB7
VCC_5.0
D2
1N914
R8
10
AC-162BYILY-H
ETH_MOSI
ETH_MOSI2
R9
2.2k
R10
3.3k
ETH_SCK
ETH_SCK2
R11
2.2k
R12
3.3k
ETH_CS
ETH_CS2
R13
2.2k
R14
3.3k
ETH_RESET
ETH_RESET2
R21
2.2k
R22
3.3k
GND
level shifters
U9A
A
U9B
B
ETH_MISO
INOUT
ETH_MISO2
U9C
C
U9D
D
U9E
E
U9F
F
U9G
VCC_5.0
VCC
GND
14
7
GND
74HCT04

Programmer Interface

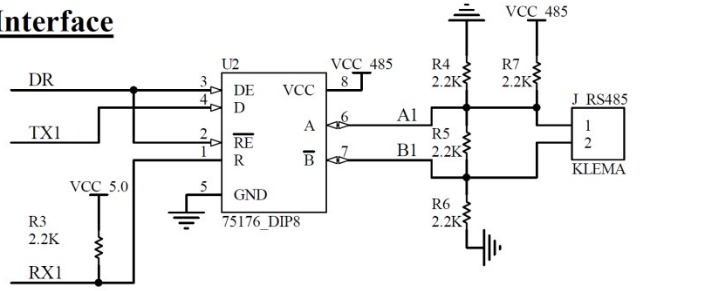


RS232 Interface

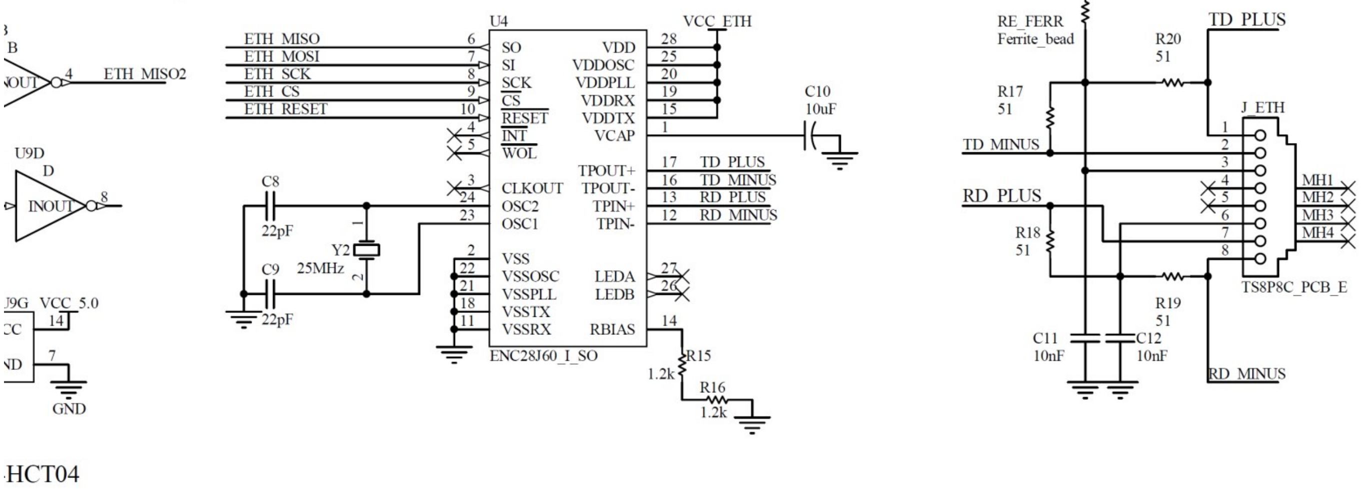




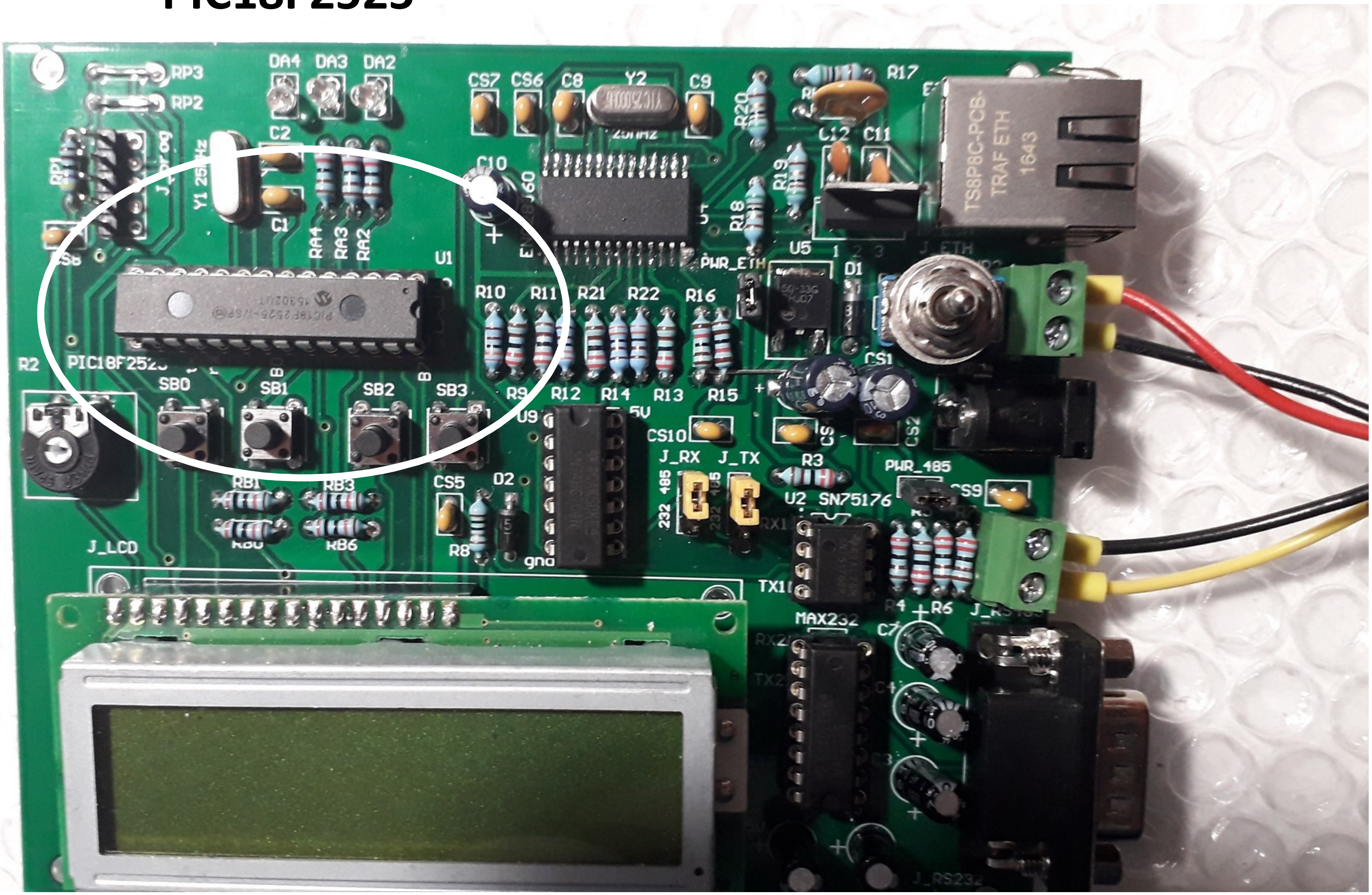
RS485 Interface

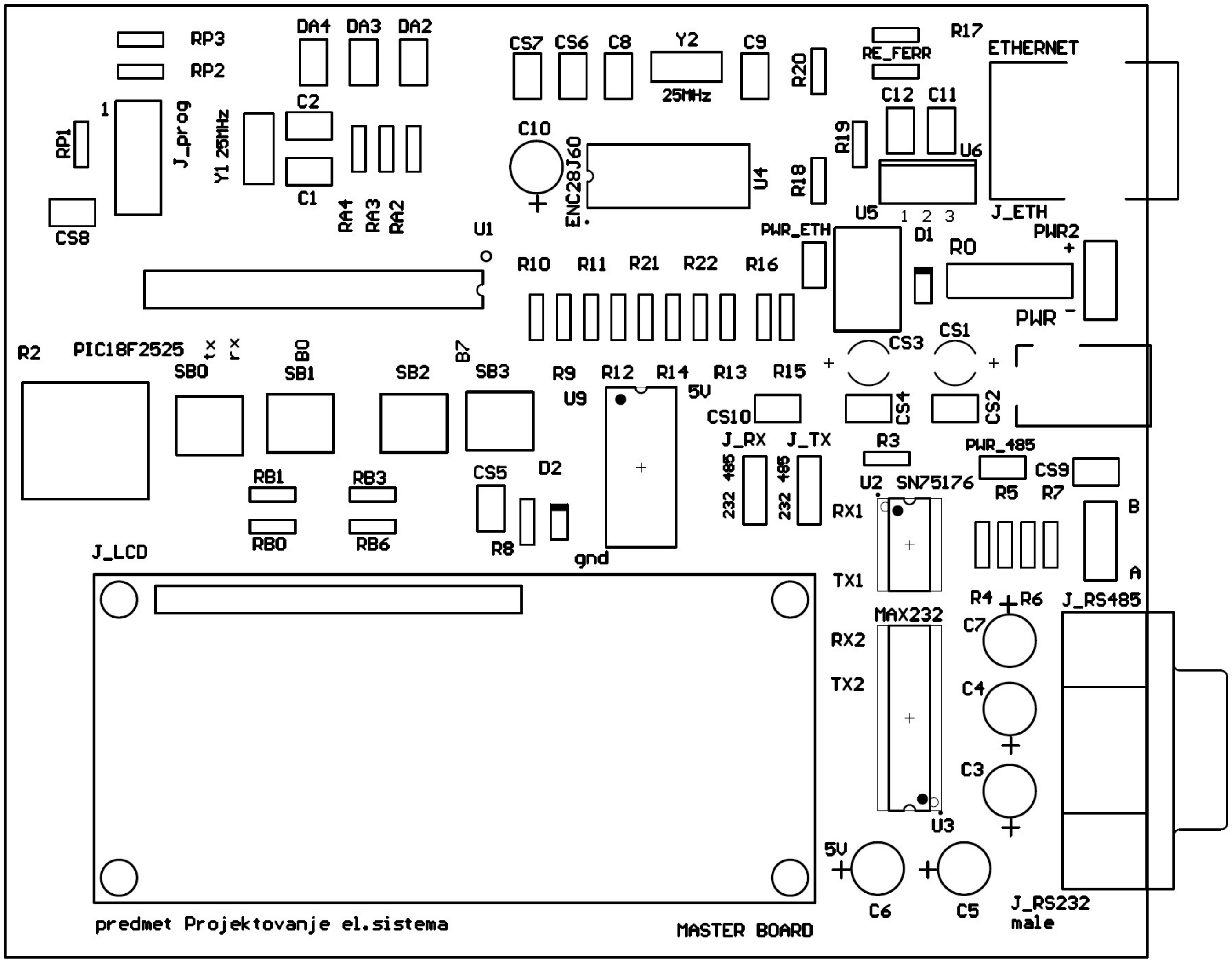


Ethernet Interface



PIC18F2525


RP3
RP2
RP1
J_prog
DA4
DA3
DA2
C2
C1
Y1 25MHz
RA4
RA3
RA2
CS8
U1
PIC18F2525
R2
SB0
SB1
SB2
SB3
RB1
RB3
RB0
RB6
J_LCD
CS7
CS6
C8
Y2
25MHz
C9
C10
R20
R19
R18
R17
C12
C11
U5
PWR_ETH
J_ETH
TS8P8C-PCB-TRAF ETH 1643
D1
R10
R11
R21
R22
R16
R9
R12
R14
R13
R15
CS1
U9
5V
CS10
CS5
D2
R8
gnd
J_RX
J_TX
232
485
R3
PWR_485
CS9
U2
SN75176
RX1
TX1
R4
R6
MAX232
RX2
TX2
C7
C4
C3
J_RS232

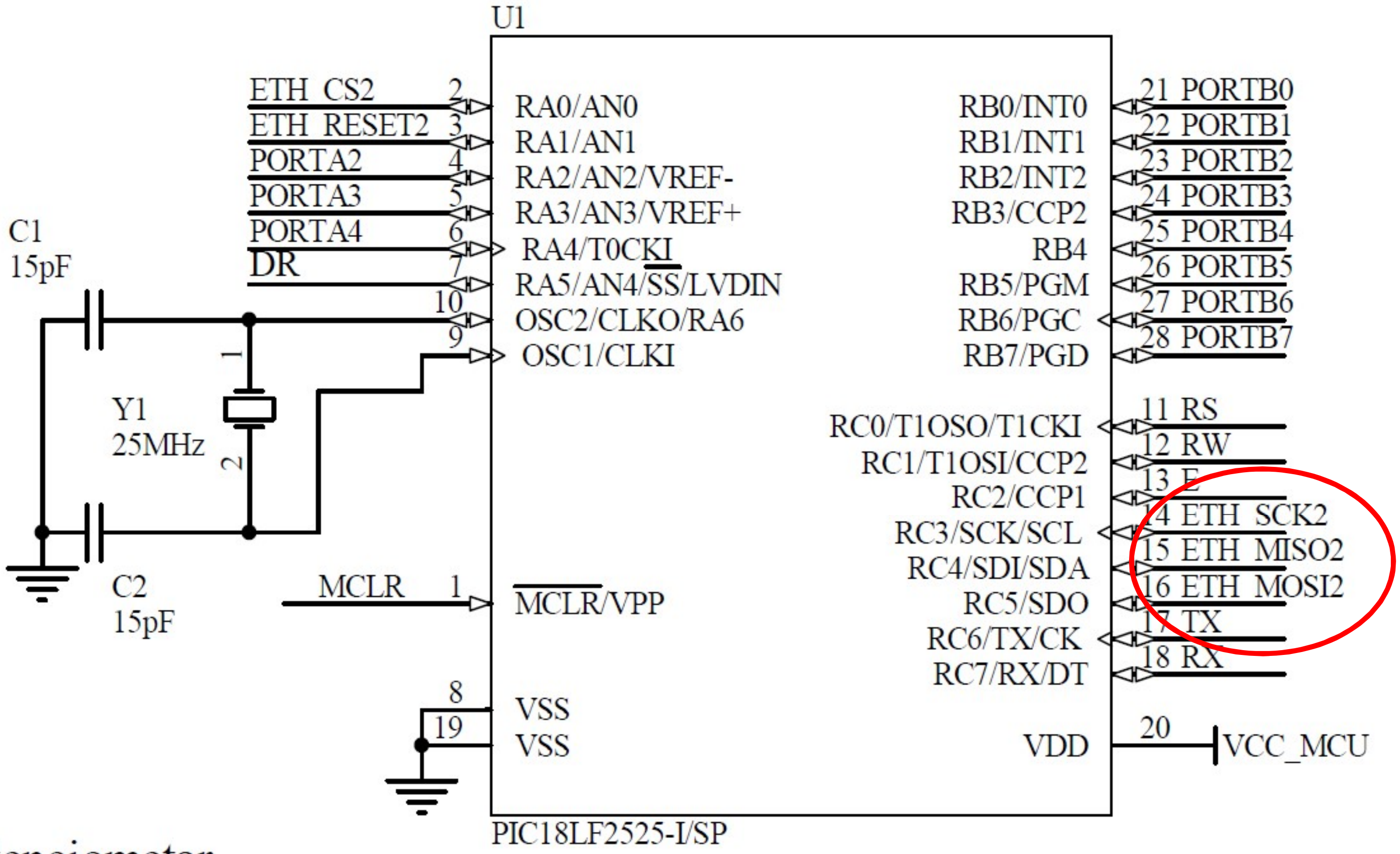

RP3
RP2
DA4
DA3
DA2
CS7
CS6
C8
Y2
25MHz
C9
R20
RE_FERR
R17
ETHERNET
C12
C11
R19
U6
R18
U5
1 2 3
J_ETH
1
RP1
J_prog
Y1 25MHz
C2
C1
RA4
RA3
RA2
CS8
C10
ENC28J60
U4
U1
PWR_ETH
D1
R0
PWR2
+
-
PWR
R10
R11
R21
R22
R16
R2
PIC18F2525
tx
rx
B0
B7
SB0
SB1
SB2
SB3
R9
R12
R14
R13
R15
CS3
CS1
U9
5V
CS10
CS4
CS2
J_RX
J_TX
R3
PWR_485
CS9
232 485
232 485
U2
SN75176
R5
R7
B
A
RB1
RB3
RB0
RB6
CS5
D2
R8
gnd
RX1
TX1
R4
R6
J_RS485
J_LCD
C7
MAX232
RX2
TX2
C4
C3
U3
5V
C6
C5
J_RS232
male
predmet Projektovanje el.sistema
MASTER BOARD

```
unsigned int   putConstString(const char *s)   {
        unsigned int ctr = 0 ;
        while(*s) {
                SPI_Ethernet_putByte(*s++) ;
                ctr++ ;
         }
        return(ctr) ;
}

unsigned int    putString (char *s) {
        unsigned int ctr = 0 ;
        while(*s){
                SPI_Ethernet_putByte(*s++) ;
                ctr++ ;
        }
        return(ctr) ;
}
```

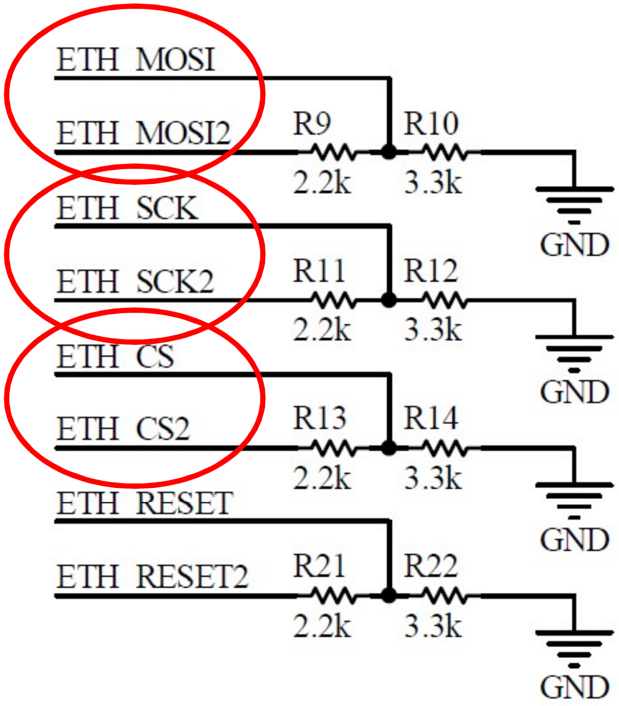
```
void dodajUNiz(char * p_ch){

    while ((*p_ch)!= 0x00) {
         niz[br_ch]= *p_ch;
         br_ch++;
         p_ch++;
    }

}
```

```
void formirajNiz() {
  unsigned char i = 0;
  char txt[4];
  br_ch = 0; // pozicioniranje na pocetak niza
  for (i = 0; i < 16; i++) {
    if (Comm[i] == 1){
      dodajUNiz("Ramp:");
      ByteToStr(i, txt);
      dodajUNiz(txt); // ID broj
      switch (Cmd[i]) { // moze biti: NO:CARDS, IDLE, VEHICLE, TIME SET
      case 0:
        dodajUNiz(" IDLE \n\n");
        break;
      case 1:
        dodajUNiz(" VEHICLE ");
        break;
      case 2:
        dodajUNiz(" TIME SET \n\n");
        break;
      case 3:
        dodajUNiz(" NO CARDS \n\n");
        break;
      default:
        break;
      }
```

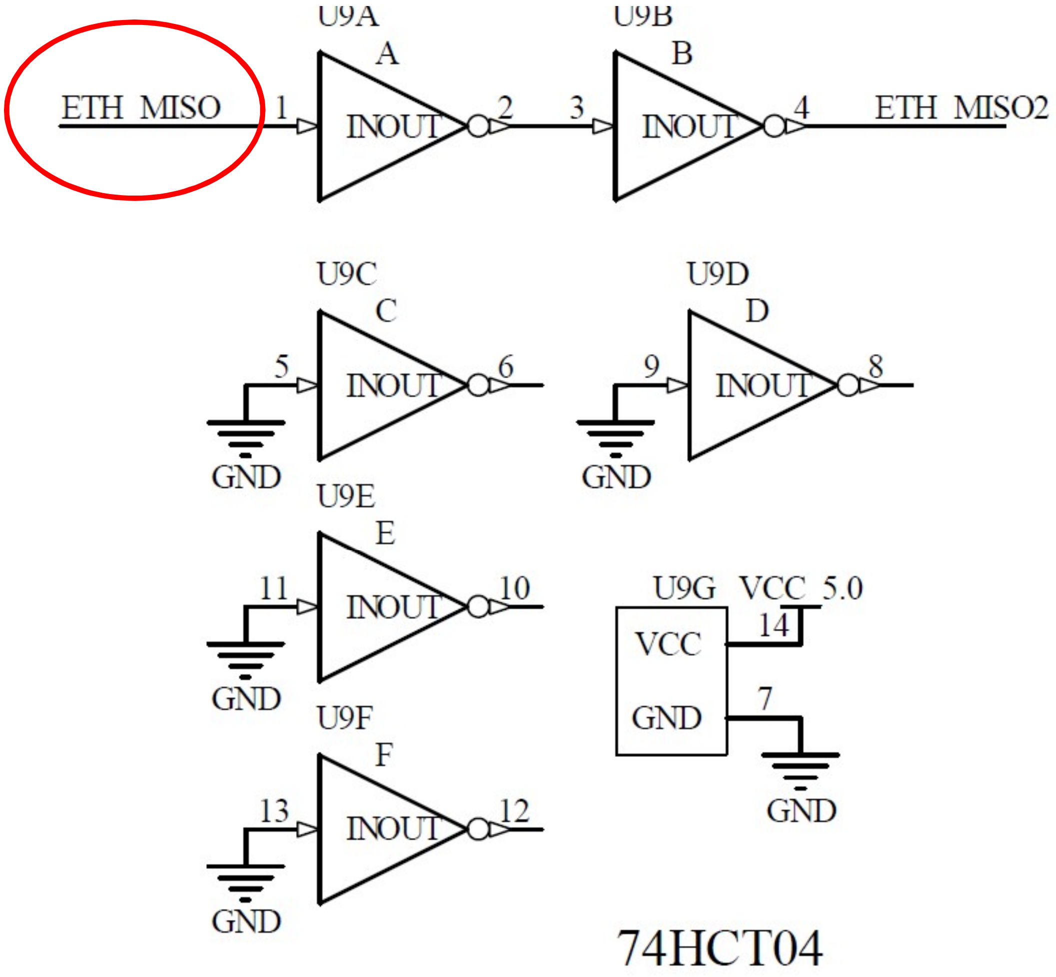
```
if (Cmd[i] == 1)
 { // VEHICLE
          pom_nula = 0x30;  // asci 0
          pom_ch = Hour[i]; // ubacivanje sati
          pom_des = (pom_ch >> 4) + pom_nula;
          pom_jed = (pom_ch & 0x0F) + pom_nula;
          niz[br_ch] = pom_des;  br_ch++;
          niz[br_ch] = pom_jed;   br_ch++;
          niz[br_ch] = ':‘;  br_ch++;

          pom_ch = Min[i]; // ubacivanje minuta
          pom_des = (pom_ch >> 4) + pom_nula;
          pom_jed = (pom_ch & 0x0F) + pom_nula;
          niz[br_ch] = pom_des;   br_ch++;
          niz[br_ch] = pom_jed;    br_ch++;
          niz[br_ch] = ':‘;  br_ch++;
          pom_ch = Sec[i]; // ubacivanje sekundi
          pom_des = (pom_ch >> 4) + pom_nula;
          pom_jed = (pom_ch & 0x0F) + pom_nula;
          niz[br_ch] = pom_des;   br_ch++;
          niz[br_ch] = pom_jed;    br_ch++;
          niz[br_ch] = ' ‘;  br_ch++;

          pom_ch = Cat[i]; // ubacivanje kategorije vozila
          pom_des = 'K';
          pom_jed = (pom_ch & 0x0F) + pom_nula;
          niz[br_ch] = pom_des;   br_ch++;
          niz[br_ch] = pom_jed;    br_ch++;
          niz[br_ch] = '\n‘;    br_ch++;
          niz[br_ch] = '\n‘;   br_ch++;
        }  // od if
      }   // od if
    }      // od for
    niz[br_ch] = 0x00;
    br_ch++; // kraj stringa
  }
```



tenciometar

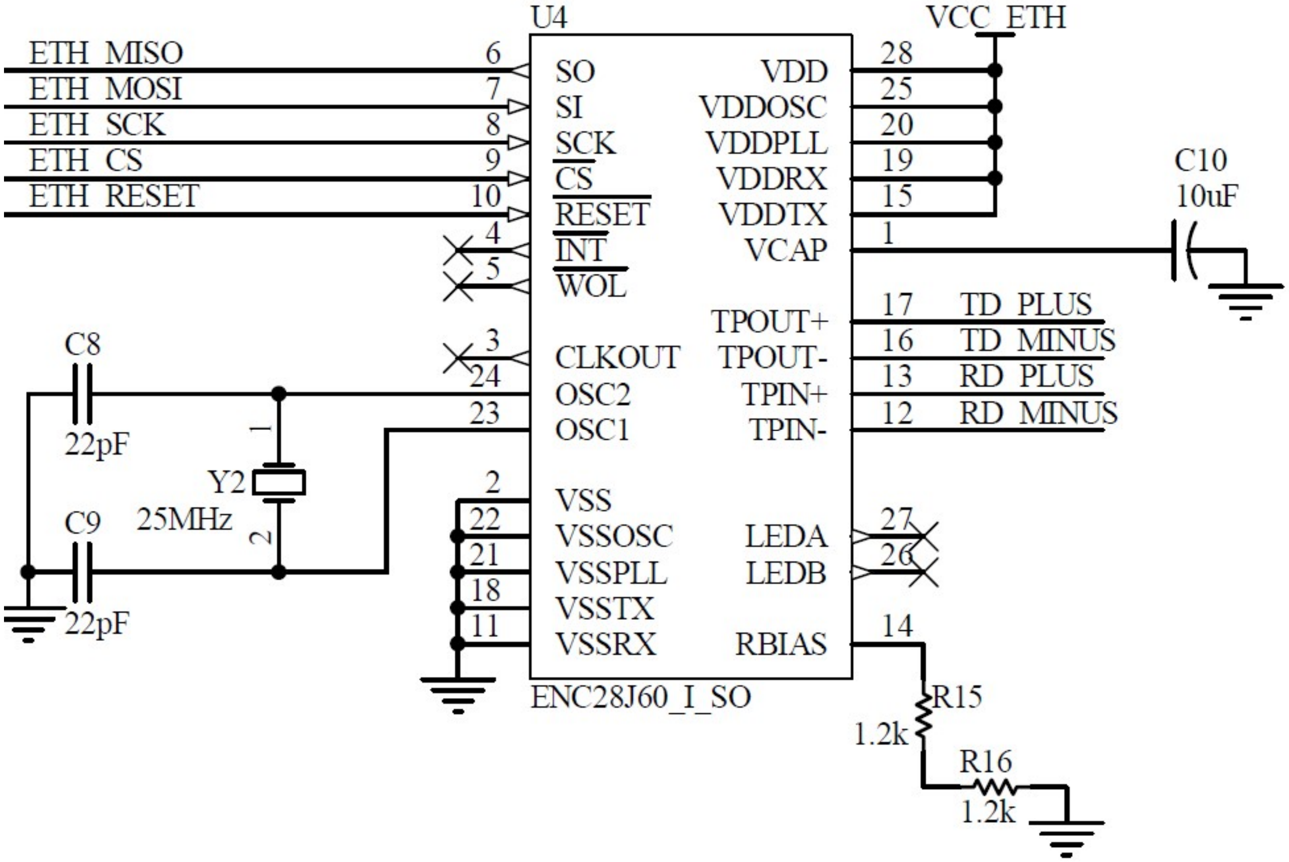


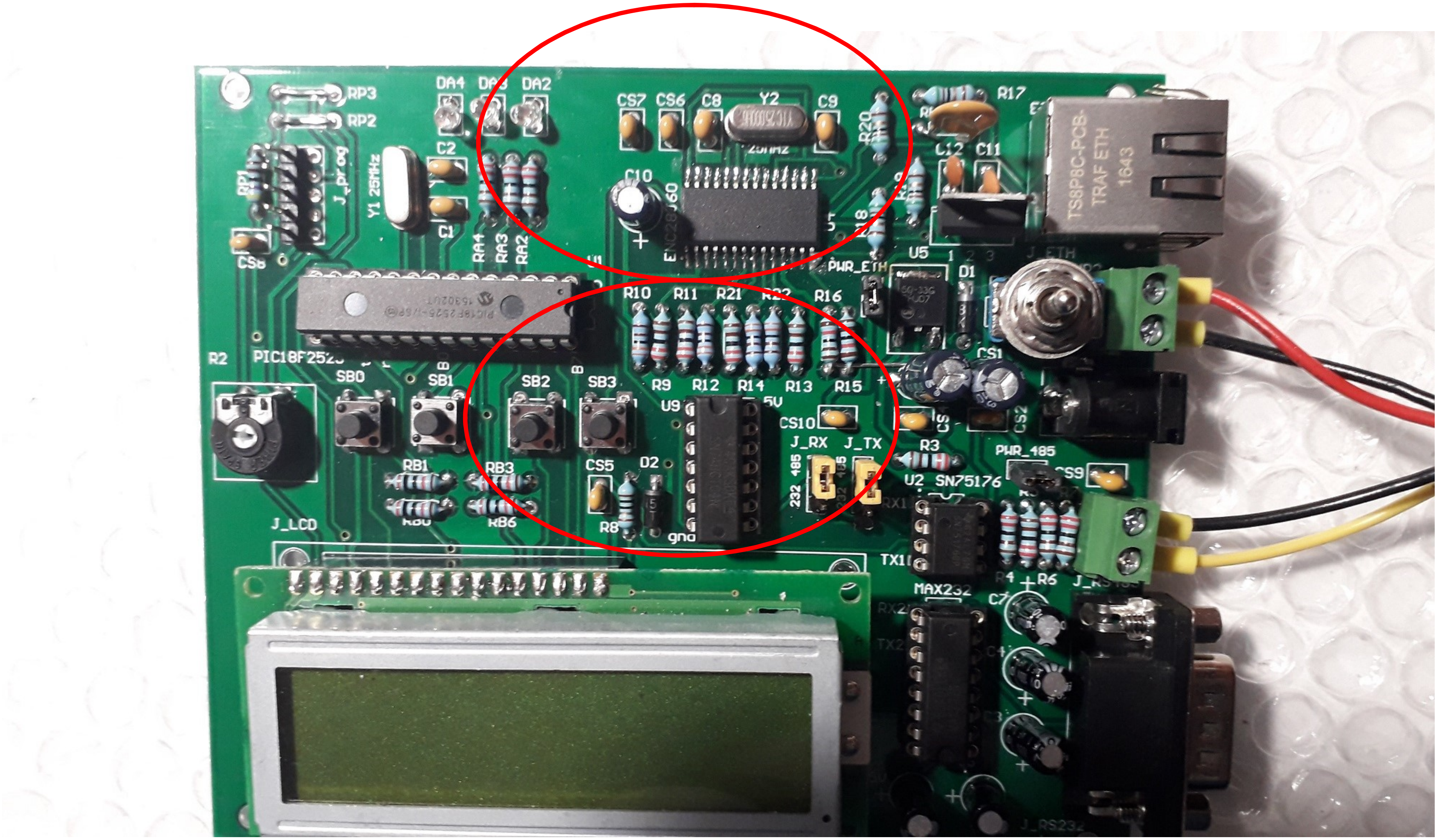
level shifters



2.2K

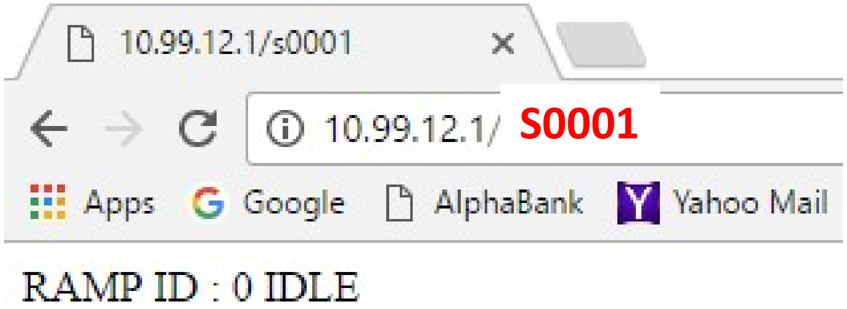
Ethernet Interface




RP3
RP2
RP1
J_prog
Y1 25MHz
DA4
DA3
DA2
C2
C1
CS8
RA4
RA3
RA2
CS7
CS6
C8
Y2
25MHz
C9
R20
R17
C10
ENC28J60
C12
C11
TS8P8C-PCB-
TRAF ETH
1643
R18
U5
D1
PWR_ETH
J_ETH
U1
PIC18F2525
R2
SB0
SB1
SB2
SB3
R10
R11
R21
R22
R16
R9
R12
R14
R13
R15
CS1
U9
5V
CS10
CS3
CS2
J_RX
J_TX
232
485
R3
PWR_485
CS9
U2
SN75176
RX1
TX1
RB1
RB3
RB6
CS5
D2
R8
gnd
J_LCD
R4
R6
MAX232
C7
RX2
TX2
C4
C3
J_RS232

Komunikacija u sistemu između CRS-a i master automata (AM) :

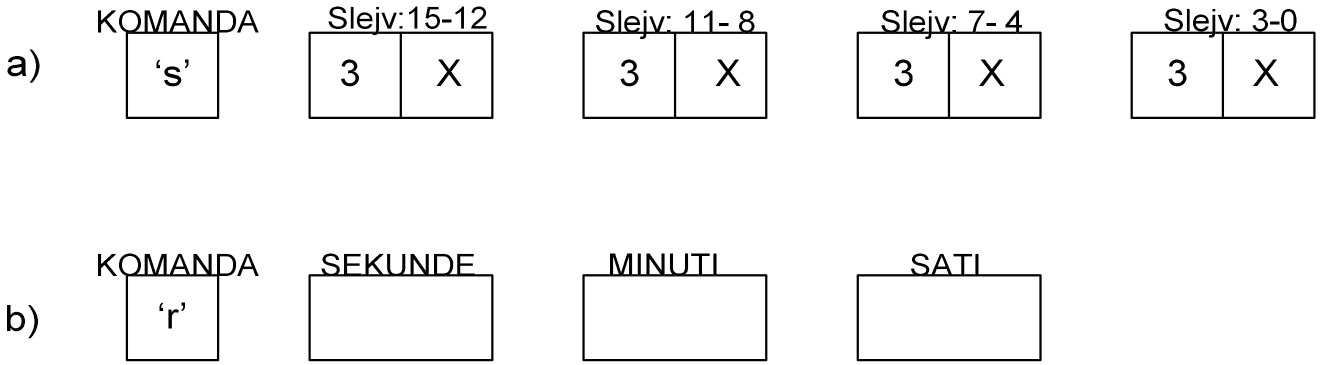
- U komandoj liniji *WEB browsera unese se* *http://XXX.XXX.XXX.XXX/komanda* (gde je XXX.XXX.XXX.XXX. IP adresa Master automata, a *komanda* predstavlja niz bajtova
- "sZZZZ" čime se zahteva status svih ulaznih rampi (preko ZZZZ podešava koje su rampe u funkciji, na primer ako je ZZZZ="0000" nijedna rampa nije uključena, ako je ZZZZ="0001" uključena je samo prva rampa, ako je ZZZZ="00??" znači da je uključeno prvih osam rampi



RAMP ID : 0 IDLE

primer “s0001” – samo Slejv sa ID brojem 0 je operativan

xϵ{0,1), kontrola slejva				15	14	13	12	xϵ{0,1), kontrola slejva				11	10	9	8	xϵ{0,1), kontrola slejva				7	6	5	4	xϵ{0,1), kontrola slejva				3	2	1	0
0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	1



a) KOMANDA ‘s’ | Slejv:15-12 3 X | Slejv: 11- 8 3 X | Slejv: 7- 4 3 X | Slejv: 3-0 3 X

b) KOMANDA ‘r’ | SEKUNDE | MINUTI | SATI

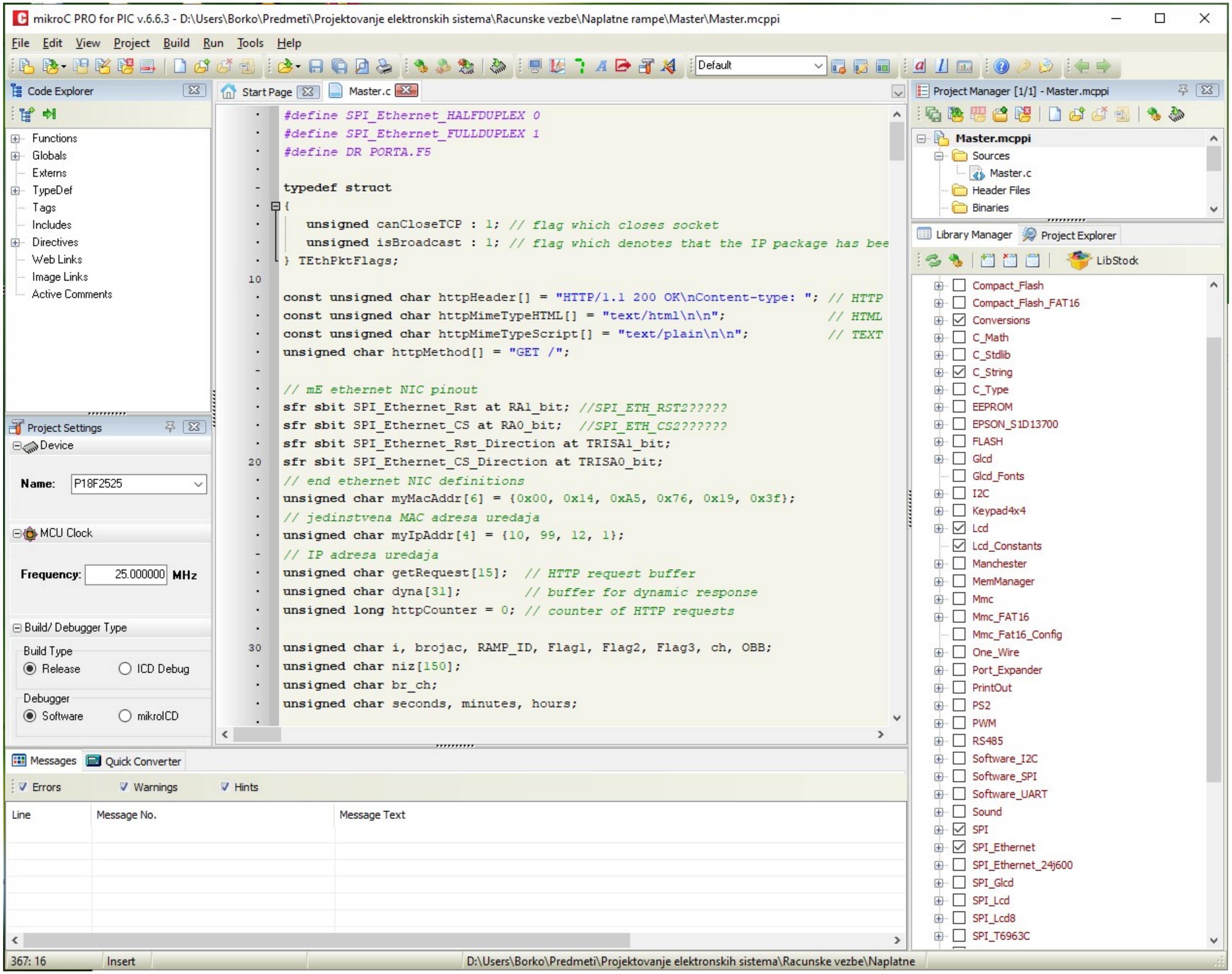
- “rZZZ” kojom podešavamo sat svim rampama (ZZZ predstavlja sekunde, minute i sate kodirane u ASCII sistemu sa ofsetom 0x30, na primer ako je ZZZ=99k onda postavljamo vreme na 9 sekundi, 9 minuta i 23 sati).

```
unsigned int    SPI_Ethernet_UserTCP (unsigned char *remoteHost, unsigned int remotePort, unsigned
int localPort, unsigned int reqLength, char *canClose) {

      unsigned int len = 0;       // my reply length
      unsigned int i;                   // general purpose integer
      if(localPort != 80) return(0);
      PORTA.F4=1;
       // get 10 first bytes only of the request, the rest does not matter here
      for(i = 0 ; i < 10 ; i++)
         getRequest[i] = SPI_Ethernet_getByte();
      getRequest[i] = 0;

      if(memcmp(getRequest, httpMethod, 5)) return(0);
       // only GET method is supported here

      if (getRequest[5]== 's') { //primio komandu za prozivanje slejvova "s"
        // s 0x3? 0x3? 0x3? 0x3?
        // 0x3? predstavlja ASCII karakter, (?  -  0, 1, 2, … , 9, A, B, C, D, E i F)
        if (((getRequest[6]& 0xF0)==0x30) && ((getRequest[7]& 0xF0)==0x30) &&
           ((getRequest[8]& 0xF0)==0x30) && ((getRequest[9]& 0xF0)==0x30)) {

           for (i=0; i<16; i++) Operation[i]=0x00;
```

```
    if ((getRequest[6]&0x08)==0x08) Operation[15] = 0x01;
    if ((getRequest[6]&0x04)==0x04) Operation[14] = 0x01;
    if ((getRequest[6]&0x02)==0x02) Operation[13] = 0x01;
    if ((getRequest[6]&0x01)==0x01) Operation[12] = 0x01;
    if ((getRequest[7]&0x08)==0x08) Operation[11] = 0x01;
    if ((getRequest[7]&0x04)==0x04) Operation[10] = 0x01;
    if ((getRequest[7]&0x02)==0x02) Operation[9] = 0x01;
    if ((getRequest[7]&0x01)==0x01) Operation[8] = 0x01;
    if ((getRequest[8]&0x08)==0x08) Operation[7] = 0x01;
    if ((getRequest[8]&0x04)==0x04) Operation[6] = 0x01;
    if ((getRequest[8]&0x02)==0x02) Operation[5] = 0x01;
    if ((getRequest[8]&0x01)==0x01) Operation[4] = 0x01;
    if ((getRequest[9]&0x08)==0x08) Operation[3] = 0x01;
    if ((getRequest[9]&0x04)==0x04) Operation[2] = 0x01;
    if ((getRequest[9]&0x02)==0x02) Operation[1] = 0x01;
    if ((getRequest[9]&0x01)==0x01) Operation[0] = 0x01;
    //postaviPortove();
  } //  if (((getRequest[6]& 0xF0) ....
} // if (getRequest[5]==0x73)
```

```
    if (getRequest[5]== ‘r') {
        // primio komandu za podesavanje RTC  "r"
        Flag2=0x01;
        // podiže FLAG 2 za postavljanje sata realnog vremena
        seconds=getRequest[6];  //nova vrednost za sekunde
        minutes=getRequest[7]; // nova vrednost za minute
        hours=getRequest[8];  // nova vrednost za sate
     }
     if(len == 0)  {
        FormirajNiz();
        len =  putConstString(httpHeader) ;             // HTTP header
        len += putConstString(httpMimeTypeHTML) ;
        len += putString(niz);      // with HTML MIME type
        for (i=0;i<16;i++){   // inicijalizacija
           Comm[i]=0x00;
           Cmd[i]=0x00;
           Cat[i]=0x00;
           Hour[i]=0x00;
           Min[i]=0x00;
           Sec[i]=0x00;
        }
     } //  if(len == 0)
     return(len) ;  // return to the library with the number of bytes to transmit
}
```

```
unsigned int SPI_Ethernet_UserUDP(unsigned char *remoteHost, unsigned int
      remotePort, unsigned int destPort, unsigned int reqLength,  TEthPktFlags * flags) {
    return 0 ;
}
```



mikroC PRO for PIC v.6.6.3 - D:\Users\Borko\Predmeti\Projektovanje elektronskih sistema\Racunske vezbe\Naplatne rampe\Master\Master.mcppi

```
#define SPI_Ethernet_HALFDUPLEX 0
#define SPI_Ethernet_FULLDUPLEX 1
#define DR PORTA.F5

typedef struct
{
   unsigned canCloseTCP : 1; // flag which closes socket
   unsigned isBroadcast : 1; // flag which denotes that the IP package has bee
} TEthPktFlags;

const unsigned char httpHeader[] = "HTTP/1.1 200 OK\nContent-type: "; // HTTP
const unsigned char httpMimeTypeHTML[] = "text/html\n\n";             // HTML
const unsigned char httpMimeTypeScript[] = "text/plain\n\n";          // TEXT
unsigned char httpMethod[] = "GET /";

// mE ethernet NIC pinout
sfr sbit SPI_Ethernet_Rst at RA1_bit; //SPI_ETH_RST2?????
sfr sbit SPI_Ethernet_CS at RA0_bit;  //SPI_ETH_CS2??????
sfr sbit SPI_Ethernet_Rst_Direction at TRISA1_bit;
sfr sbit SPI_Ethernet_CS_Direction at TRISA0_bit;
// end ethernet NIC definitions
unsigned char myMacAddr[6] = {0x00, 0x14, 0xA5, 0x76, 0x19, 0x3f};
// jedinstvena MAC adresa uredaja
unsigned char myIpAddr[4] = {10, 99, 12, 1};
// IP adresa uredaja
unsigned char getRequest[15];  // HTTP request buffer
unsigned char dyna[31];        // buffer for dynamic response
unsigned long httpCounter = 0; // counter of HTTP requests

unsigned char i, brojac, RAMP_ID, Flag1, Flag2, Flag3, ch, OBB;
unsigned char niz[150];
unsigned char br_ch;
unsigned char seconds, minutes, hours;
```

Device

Name: P18F2525

MCU Clock

Frequency: 25.000000 **MHz**

Build/ Debugger Type

Build Type

● Release ○ ICD Debug

Debugger

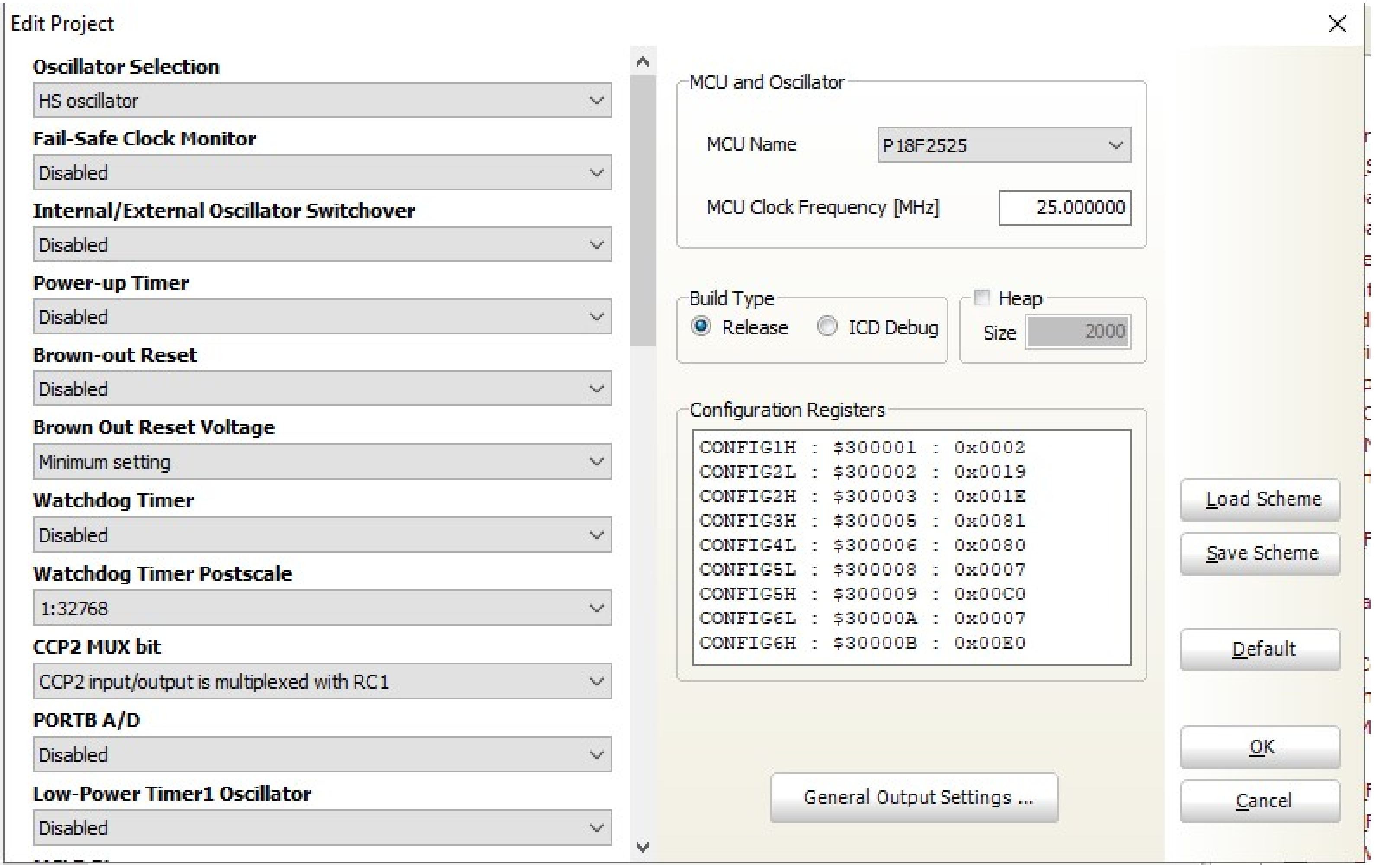
● Software ○ mikroICD

☑ Errors ☑ Warnings ☑ Hints

Line	Message No.	Message Text

☐ Compact_Flash
☐ Compact_Flash_FAT16
☑ Conversions
☐ C_Math
☐ C_Stdlib
☑ C_String
☐ C_Type
☐ EEPROM
☐ EPSON_S1D13700
☐ FLASH
☐ Glcd
☐ Glcd_Fonts
☐ I2C
☐ Keypad4x4
☑ Lcd
☑ Lcd_Constants
☐ Manchester
☐ MemManager
☐ Mmc
☐ Mmc_FAT16
☐ Mmc_Fat16_Config
☐ One_Wire
☐ Port_Expander
☐ PrintOut
☐ PS2
☐ PWM
☐ RS485
☐ Software_I2C
☐ Software_SPI
☐ Software_UART
☐ Sound
☑ SPI
☑ SPI_Ethernet
☐ SPI_Ethernet_24j600
☐ SPI_Glcd
☐ SPI_Lcd
☐ SPI_Lcd8
☐ SPI_T6963C

367: 16 Insert D:\Users\Borko\Predmeti\Projektovanje elektronskih sistema\Racunske vezbe\Naplatne



Oscillator Selection
HS oscillator

Fail-Safe Clock Monitor
Disabled

Internal/External Oscillator Switchover
Disabled

Power-up Timer
Disabled

Brown-out Reset
Disabled

Brown Out Reset Voltage
Minimum setting

Watchdog Timer
Disabled

Watchdog Timer Postscale
1:32768

CCP2 MUX bit
CCP2 input/output is multiplexed with RC1

PORTB A/D
Disabled

Low-Power Timer1 Oscillator
Disabled

MCU and Oscillator

MCU Name: P18F2525

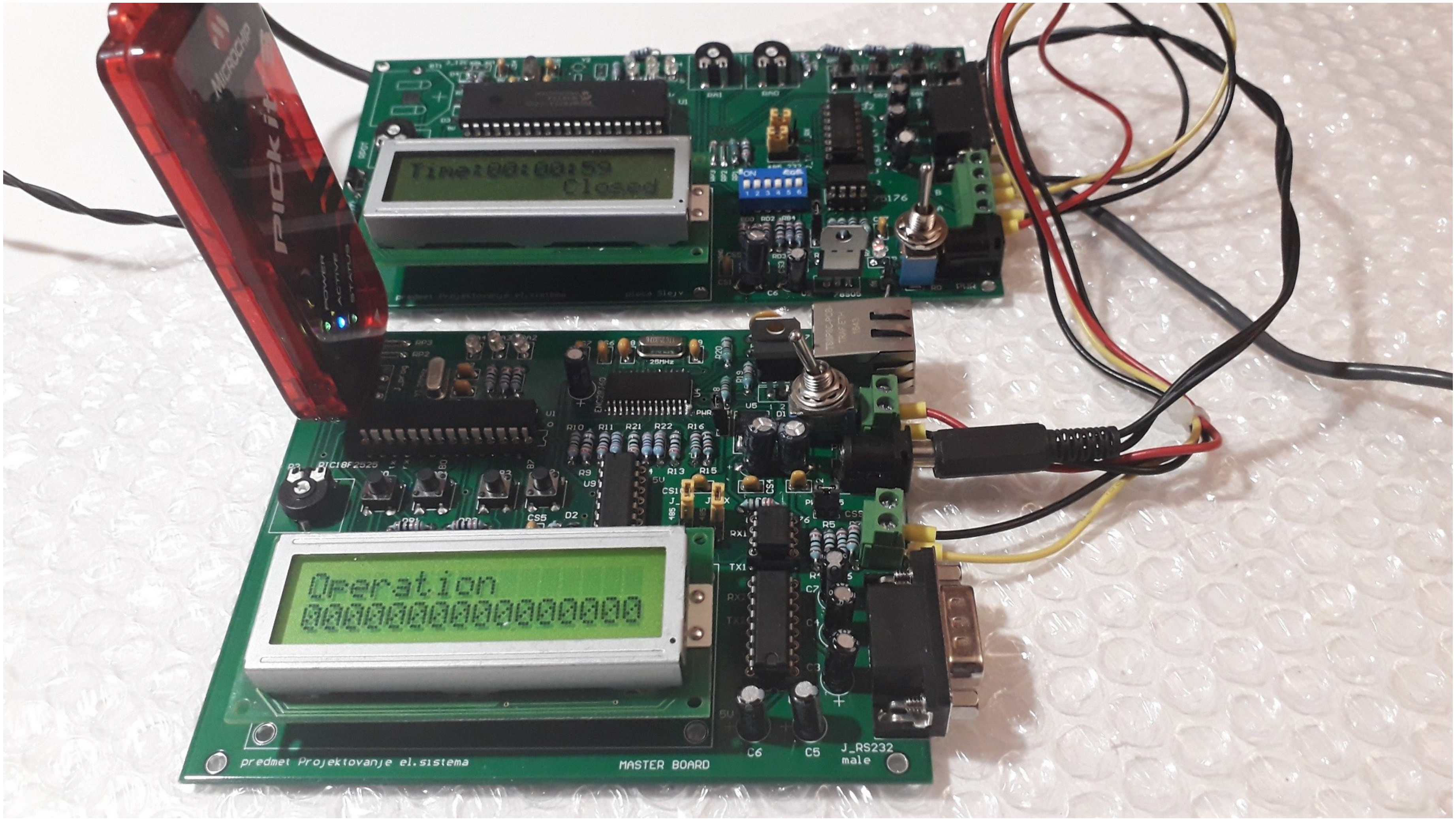
MCU Clock Frequency [MHz]: 25.000000

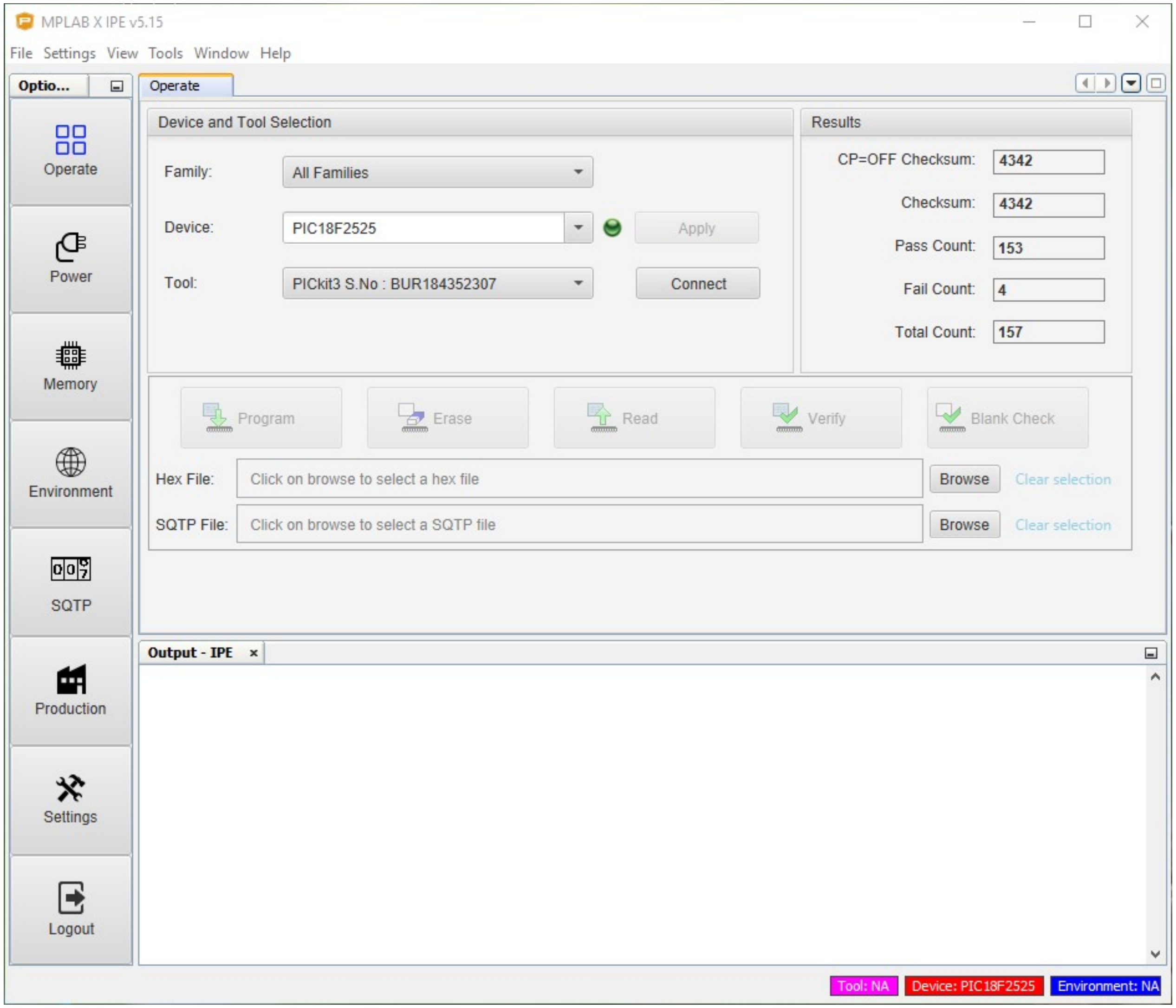
Build Type: ● Release ○ ICD Debug

□ Heap Size: 2000

Configuration Registers

```
CONFIG1H : $300001 : 0x0002
CONFIG2L : $300002 : 0x0019
CONFIG2H : $300003 : 0x001E
CONFIG3H : $300005 : 0x0081
CONFIG4L : $300006 : 0x0080
CONFIG5L : $300008 : 0x0007
CONFIG5H : $300009 : 0x00C0
CONFIG6L : $30000A : 0x0007
CONFIG6H : $30000B : 0x00E0
```


PICkit
MICROCHIP
Time:00:00:59
Closed
Operation
0000000000000000
PIC18F2525
predmet Projektovanje el.sistema
MASTER BOARD
J_RS232
male
25MHz
C6
C5



Family: All Families

Device: PIC18F2525

Tool: PICkit3 S.No : BUR184352307

CP=OFF Checksum: 4342

Checksum: 4342

Pass Count: 153

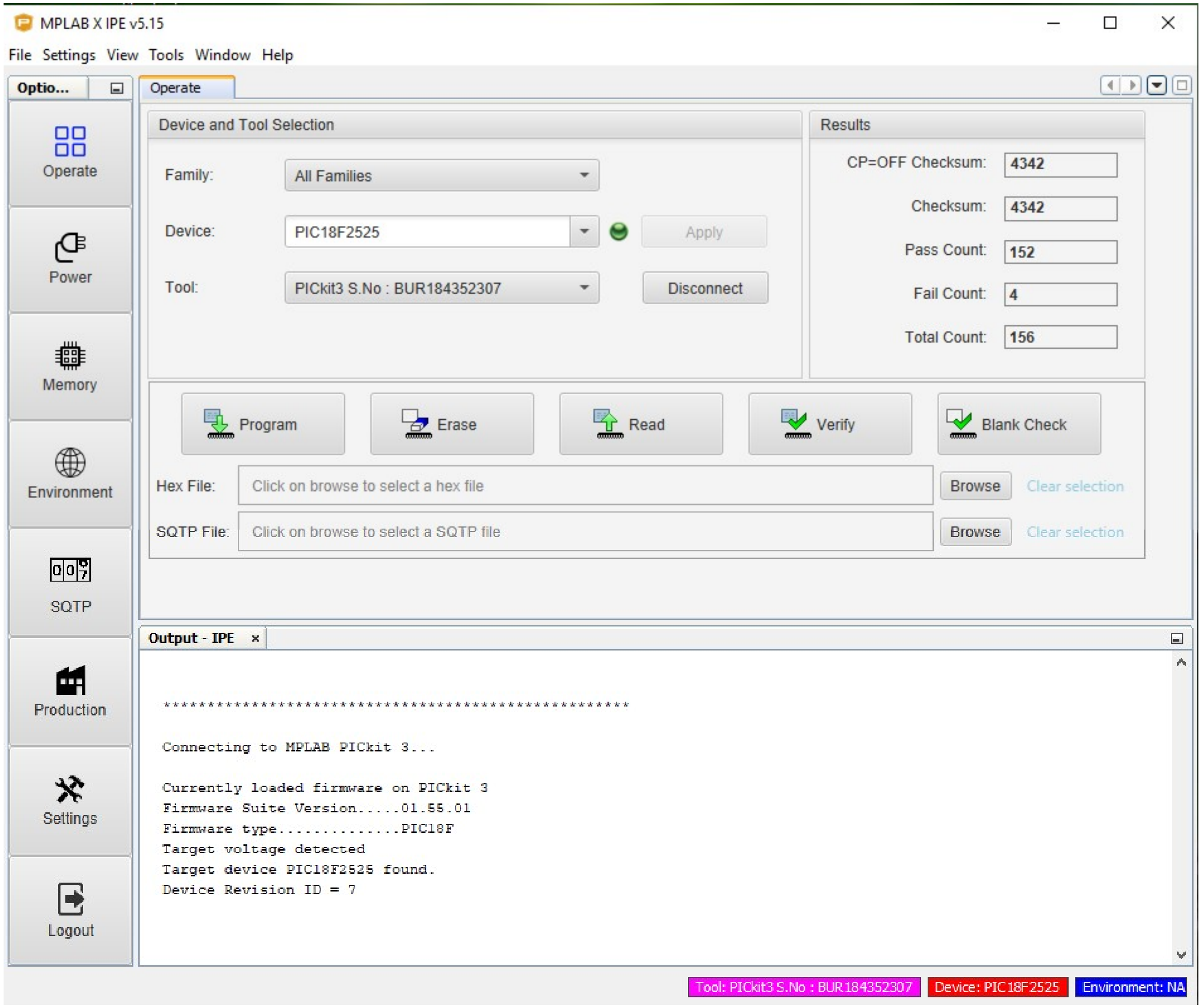
Fail Count: 4

Total Count: 157

Hex File: Click on browse to select a hex file

SQTP File: Click on browse to select a SQTP file

Tool: NA Device: PIC18F2525 Environment: NA



Family: All Families

Device: PIC18F2525

Tool: PICkit3 S.No : BUR184352307

CP=OFF Checksum: 4342

Checksum: 4342

Pass Count: 152

Fail Count: 4

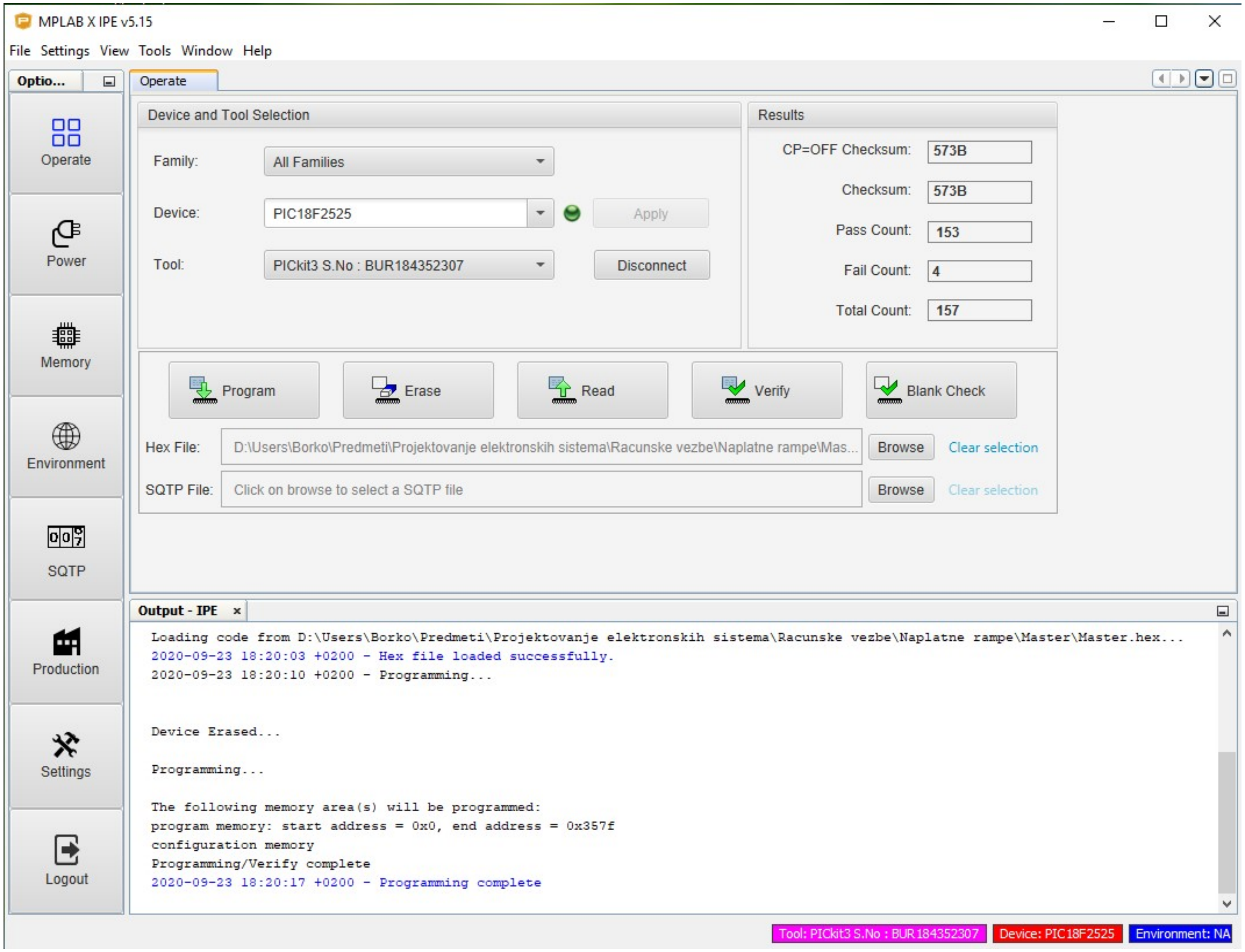
Total Count: 156

Hex File: Click on browse to select a hex file

SQTP File: Click on browse to select a SQTP file

```
******************************************************

Connecting to MPLAB PICkit 3...

Currently loaded firmware on PICkit 3
Firmware Suite Version.....01.55.01
Firmware type..............PIC18F
Target voltage detected
Target device PIC18F2525 found.
Device Revision ID = 7
```

Tool: PICkit3 S.No : BUR184352307 Device: PIC18F2525 Environment: NA



Family: All Families

Device: PIC18F2525

Tool: PICkit3 S.No : BUR184352307

CP=OFF Checksum: 573B

Checksum: 573B

Pass Count: 153

Fail Count: 4

Total Count: 157

Hex File: D:\Users\Borko\Predmeti\Projektovanje elektronskih sistema\Racunske vezbe\Naplatne rampe\Mas...

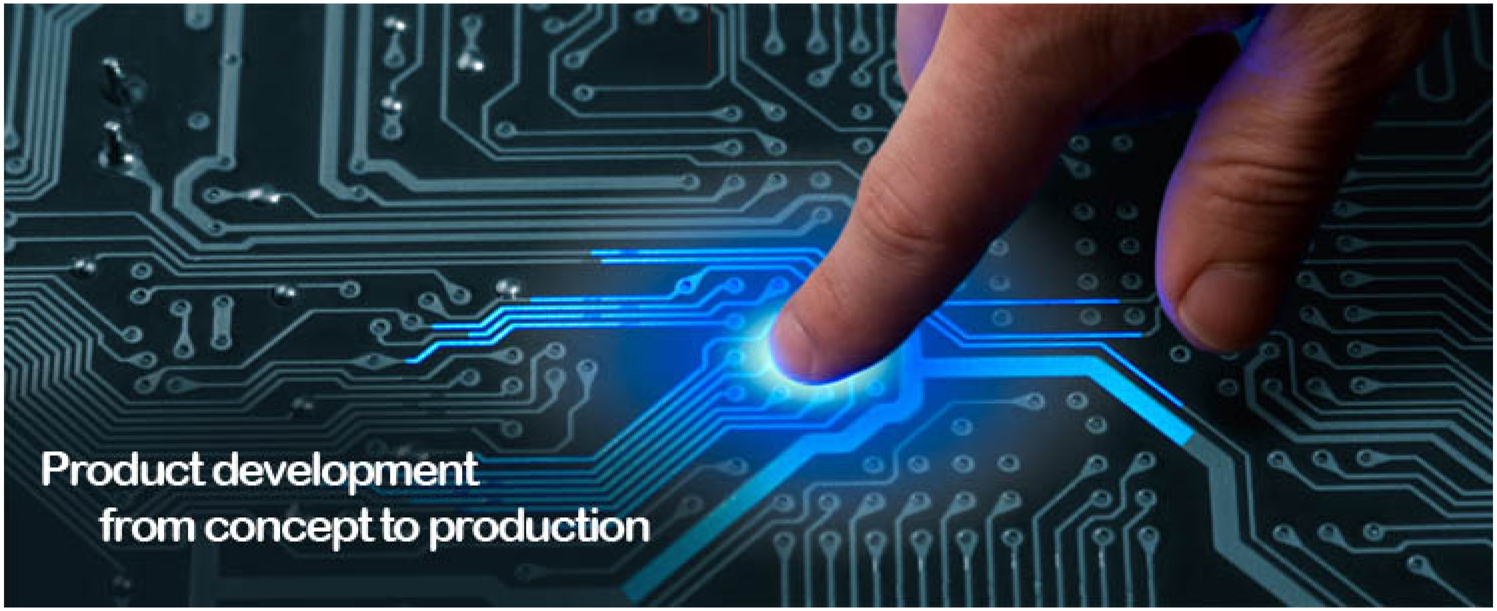
SQTP File: Click on browse to select a SQTP file

```
Loading code from D:\Users\Borko\Predmeti\Projektovanje elektronskih sistema\Racunske vezbe\Naplatne rampe\Master\Master.hex...
2020-09-23 18:20:03 +0200 - Hex file loaded successfully.
2020-09-23 18:20:10 +0200 - Programming...


Device Erased...

Programming...

The following memory area(s) will be programmed:
program memory: start address = 0x0, end address = 0x357f
configuration memory
Programming/Verify complete
2020-09-23 18:20:17 +0200 - Programming complete
```

Tool: PICkit3 S.No : BUR184352307 | Device: PIC18F2525 | Environment: NA

Projektna dokumentacija – ovo treba da sadrži svaki seminarski rad

- **Opis zadatka**
- **Funkcionalna i nefunkcionalna specifikacija**
- **Arhitektura sistema**
- **Realizacija komunikacionog protokola**
- **Hardverska realizacija automata**
  - *Tabela: Raspored iskorišćenih ulaznih i izlaznih pinova mikrokontrolera*
- **Realizacija firmvera automata**
  - *Tabela najvažnijih promenljivih koji se koriste u programskom kodu*
  - *Tabela najvažnijih funkcija koji se koriste u programskom kodu*
  - *Detaljan opis najvažnijih funkcija sa segmentima koda*
- **Postupak verifikacije i vrednovanja projekta**
  - *Rezultati rada sistema*
  - *Stepen ispunjenosti korisničkih zahteva*
- **Detaljno uputstvo za upotrebu**


Product development
from concept to production