

UNIVERZITET U SARAJEVU



ELEKTROTEHNIČKI FAKULTET  
ODSJEK ZA AUTOMATIKU I ELEKTRONIKU

---

# Detektor kvalitete zraka, sa naznakom na etanolu, ugljen monoksidu i hidrogen sulfidu

---

**Baždar Faris, 18413**

*Predmet: Praktikum elektronike*

Akadska godina: 2020/2021.

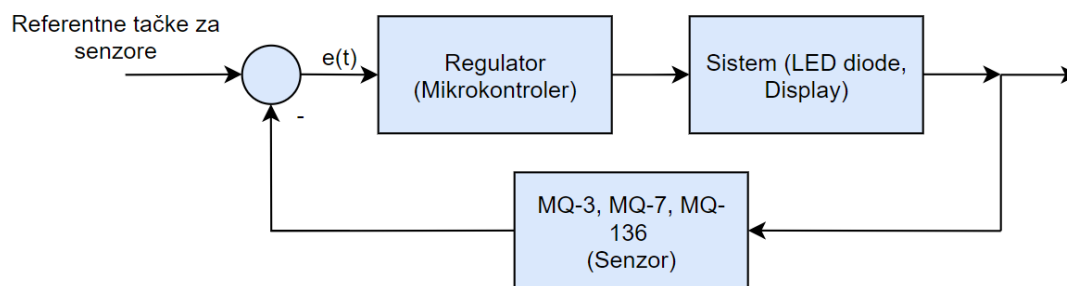
Sarajevo, Februar 2021.

# 1 Razrada zadatka

## 1.1 Uvod

Za projektni zadatak realiziran je detektor kvalitete zraka sa akcentom na 3 specifična gasa (etanol, ugljen monoksid, hidrogen sulfid). Koriste se 3 senzora sa kojih se očitava naponska vrijednost u opsegu od 0-5V. Prikaz kvalitete zraka se vrši nad 3 LED diode, gdje je korištena crvena (indikator visokog nivoa hazarda u zraku), žuta (indikator osrednjeg kvaliteta zraka) i zelena dioda (indikator kvalitetnog kvaliteta zraka). Pored toga specifične informacije sa senzora se prikazuju na LCD Displeju. Senzor za etanol prikazuje vrijednost u mg/L, senzor za ugljen monoksid i hidrogen sulfid prikazuju vrijednosti u ppm (*engl. parts per milion*). Mikrokontroler kao i senzori napajaju se sa 5V naizmjenicnog napona, pa je izvršena konverzija sa 12V koristeći regulator. Korišten je i eksterni kvarcni kristal kao oscilator.

## 1.2 Blok struktura



Slika 1: Blok struktura projektnog zadatka

## 1.3 Kod

```
1 #define _XTAL_FREQ 8000000
2
3 #include <xc.h>
4 #include <stdio.h>
5
6 #pragma config FOSC=HS, WDTE=OFF, PWRTE=OFF, MCLRE=ON, CP=OFF, CPD=OFF, BOREN=
    OFF, CLKOUTEN=OFF
7 #pragma config IESO=OFF, FCMEN=OFF, WRT=OFF, VCAPEN=OFF, PLLEN=OFF, STVREN=
    OFF, LVP=OFF
8
9 #define zelena RB0
10 #define zuta RB1
11 #define crvena RB2
12
13 #define RS RD2
14 #define EN RD3
15 #define D4 RD4
16 #define D5 RD5
17 #define D6 RD6
```

```

18 #define D7 RD7
19
20 double naponPrvi = 0;
21 double naponDrugi = 0;
22 double naponTreci = 0;
23
24 double sumaNapona = 0;
25
26 void Lcd_SetBit(char data_bit)
27 {
28     if (data_bit & 1)
29         D4 = 1;
30     else
31         D4 = 0;
32     if (data_bit & 2)
33         D5 = 1;
34     else
35         D5 = 0;
36     if (data_bit & 4)
37         D6 = 1;
38     else
39         D6 = 0;
40     if (data_bit & 8)
41         D7 = 1;
42     else
43         D7 = 0;
44 }
45
46 void Lcd_Cmd(char a) {
47     RS = 0;
48     Lcd_SetBit(a);
49     EN = 1;
50     __delay_ms(4);
51     EN = 0;
52 }
53
54 Lcd_Clear() {
55     Lcd_Cmd(0);
56     Lcd_Cmd(1);
57 }
58
59 void Lcd_Set_Cursor(char a, char b) {
60     char temp, z, y;
61     if (a == 1) {
62         temp = 0x80 + b - 1;
63         z = temp >> 4;
64         y = temp & 0x0F;
65         Lcd_Cmd(z);
66         Lcd_Cmd(y);
67     } else if (a == 2) {
68         temp = 0xC0 + b - 1;
69         z = temp >> 4;
70         y = temp & 0x0F;
71         Lcd_Cmd(z);
72         Lcd_Cmd(y);
73     }
74 }
75

```

```

76 void Lcd_Start() {
77     Lcd_SetBit(0x00);
78     for (int i = 1065244; i <= 0; i--) NOP();
79     Lcd_Cmd(0x03);
80     __delay_ms(5);
81     Lcd_Cmd(0x03);
82     __delay_ms(11);
83     Lcd_Cmd(0x03);
84     Lcd_Cmd(0x02);
85     Lcd_Cmd(0x02);
86     Lcd_Cmd(0x08);
87     Lcd_Cmd(0x00);
88     Lcd_Cmd(0x0C);
89     Lcd_Cmd(0x00);
90     Lcd_Cmd(0x06);
91 }
92
93 void Lcd_Print_Char(char data)
94 {
95     char Lower_Nibble, Upper_Nibble;
96     Lower_Nibble = data & 0x0F;
97     Upper_Nibble = data & 0xF0;
98     RS = 1;
99     Lcd_SetBit(Upper_Nibble >> 4);
100    EN = 1;
101    for (int i = 2130483; i <= 0; i--) NOP();
102    EN = 0;
103    Lcd_SetBit(Lower_Nibble);
104    EN = 1;
105    for (int i = 2130483; i <= 0; i--) NOP();
106    EN = 0;
107 }
108
109 void Lcd_Print_String(char *a) {
110     int i;
111     for (i = 0; a[i] != '\0'; i++)
112         Lcd_Print_Char(a[i]);
113 }
114
115 void printToLCD(void) {
116     char text[17];
117     Lcd_Clear();
118
119     int percentage1 = naponPrvi/5.0 * 10;
120     int percentage2 = naponDrugi/5.0 * 2000;
121     int percentage3 = naponTreci/5.0 * 100;
122
123
124     sprintf(text, "ALK:%1d CO:%3d", percentage1, percentage2);
125     Lcd_Set_Cursor(1, 1);
126     Lcd_Print_String(text);
127
128
129     sprintf(text, "H2s:%3d", percentage3);
130     Lcd_Set_Cursor(2, 1);
131     Lcd_Print_String(text);
132     __delay_ms(2000);
133 }

```

```

134
135 void initLCDSeparately(void){
136     Lcd_Start();
137 }
138
139 void initLCD(void){
140     TRISD = 0x00;
141     ANSELD = 0x00;
142     LATD = 0x00;
143 }
144
145
146
147 void setupOtherPorts(void){
148     TRISBbits.TRISB0 = 0;
149     TRISBbits.TRISB1 = 0;
150     TRISBbits.TRISB2 = 0;
151     ANSELB = 0;
152     LATB = 0;
153 }
154
155 void initAD(void){
156     TRISAbits.TRISA0 = 1;
157     TRISAbits.TRISA1 = 1;
158     TRISAbits.TRISA2 = 1;
159     ANSELA = 0x07;
160
161     ADCON1bits.ADFM = 1;
162     ADCON1bits.ADCS2 = 1;
163     ADCON1bits.ADCS1 = 1;
164     ADCON1bits.ADCS0 = 1;
165
166     ADCON1bits.ADPREF0 = 0;
167     ADCON1bits.ADPREF1 = 0;
168     ADCON1bits.ADNREF = 0;
169
170     ADCON0bits.ADON = 1;
171
172     ADCON0bits.CHS4 = 0;
173     ADCON0bits.CHS3 = 0;
174     ADCON0bits.CHS2 = 0;
175     ADCON0bits.CHS1 = 0;
176     ADCON0bits.CHS0 = 0;
177 }
178
179 void pollfirst(void){
180     ADCON0bits.ADGO = 1;
181     while(ADCON0bits.ADGO);
182     naponPrvi=5/1023.0*(ADRESH*256+ADRESL);
183 }
184
185 void pollsecond(void){
186     ADCON0bits.CHS0 = 1;
187     ADCON0bits.ADGO = 1;
188     while(ADCON0bits.ADGO);
189     naponDrugi=5/1023.0*(ADRESH*256+ADRESL);
190 }
191

```

```

192 void pollthird(void){
193     ADCON0bits.CHS0 = 0;
194     ADCON0bits.CHS1 = 1;
195     ADCON0bits.ADGO = 1;
196     while(ADCON0bits.ADGO);
197     naponTreci=5/1023.0*(ADRESH*256+ADRESL);
198     ADCON0bits.CHS1 = 0;
199 }
200
201 void display(double a, double b, double c){
202     if((a>=2.5) || (b>=1) || (c>=1.5)){
203         zelena = zuta = 0;
204         crvena = 1;
205     }else if((a>=1.5 && a <2.5) || (b>=0.25 && b<1) || (c>=0.5 && c<1.5)
206 ) {
207         zelena = crvena = 0;
208         zuta = 1;
209     }else{
210         zuta = crvena = 0;
211         zelena = 1;
212     }
213 }
214 void ADDisplay(void){
215     pollfirst();
216     pollsecond();
217     pollthird();
218     display(naponPrvi, naponDrugi, naponTreci);
219 }
220
221 void main(void) {
222     setupOtherPorts();
223     initAD();
224     initLCD();
225     initLCDSeparately();
226     while(1){
227         ADDisplay();
228         printToLCD();
229     }
230     return;
231 }

```

## 1.4 Odabir senzora

Korišteni su senzori MQ-3, MQ-7, MQ-136. Svaki od tih senzora može se kalibrirati u ovisnosti u kakvoj se okolini primijenjuju. Senzori su poprilično jednostavni za korištenje ali su mnogo neprecizniji u odnosu na neke sofisticiranije senzore. Još jedna od mana ovih senzora je što je potrebno prvo ugrijati senzore kako bi mogli sa većom tačnošću da očitavaju gasove u zraku. Potrebno je poprilično duga količina vremena (preko 24h) da bi senzori očitavali sa većom tačnošću. <sup>1</sup>

---

<sup>1</sup>Datasheet-ovi se mogu pronaći na linkovima:

<https://www.sparkfun.com/datasheets/Sensors/MQ-3.pdf>

<https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf>

<http://www.sensorica.ru/pdf/MQ-136.pdf>

## 1.5 Tehničke specifikacije

### Senzor MQ-3 (Etanol)

Repsiratorni sistem apsorbira oko 62% etanola u zraku ukoliko je koncentracija od 11 do 19 mg/L, što je neovisno od količine ventilacije. Velike koncentracije etanola u zraku povećavaju promile alkohola u krvi ukoliko dođe do inhalacije. Pored toga u nekim postrojenjima jako je bitno poznavati nivo etanola (ili eventualno benzina) u nekoj prostoriji. Ukoliko se procentualno očita niže od 3mg/L alkohola u zraku, smatramo da je zrak siguran, ukoliko je iznad toga ali ispod 5mg/L, smatramo to kao osrednje zagađenje. Dok je sve iznad toga reprezentacija veoma nekvalitetnog zraka. Naš senzor mjeri od 0 do 10 mg/L pa je na taj način i zakodirano očitavanje.

### Senzor MQ-7 (Ugljen monoksid)

Na osnovu opsega senzora i istraživanja datih u fusnoti može se odrediti opseg potreban za rad sistema. (20-100) ppm jos uvijek sigurno za rad, 100-400 ppm se smatra nekvalitetnim. Sve iznad 400 se smatra potpuno nekvalitetnim, te štetnim za ljudski organizam. Na ovaj način je i zakodirano očitavanje vrijednosoti sa senzora.

### Senzor MQ-136 (Hidrogen sulfid)

(0-10) Manje promojene u metabolizmu tokom dužeg izlaganja gasu, (10-30) Iritacija u očima, Pad koncentracije (30-100) Nivo čestica u zraku koje mogu izazvati trajnu štetu po čovjekov organizam, 80-100 opasno po život. Na 100 dolazi do ozbiljnih problema u respiratnom sistemu, centralnom nervnom sistemu, kardiovaskularnom sistemu.



Slika 2: Izgled senzora koji su korišteni

MQ-3			
Symbol	Parameter name	Technical parameter	Remarks
Rs	Sensing Resistance	1MΩ - 8 MΩ (0.4mg/L alcohol )	Detecting concentration scope: 0.05mg/L—10mg/L Alcohol
α (0.4/1 mg/L)	Concentration slope rate	≤0.6	
Standard detecting condition	Temp: 20℃ ± 2℃ Humidity: 65%±5%	Vc:5V±0.1 Vh: 5V±0.1	
Preheat time	Over 24 hour		

MQ-7	symbol	Parameters	Technical parameters	Remark
	$R_s$	Surface resistance Of sensitive body	2-20k	In 100ppm Carbon Monoxide
	$\alpha$ (300/100ppm)	Concentration slope rate	Less than 0.5	$R_s$ (300ppm)/ $R_s$ (100ppm)
	Standard working condition	Temperature $-20^{\circ}\text{C} \pm 2^{\circ}\text{C}$ $V_c: 5V \pm 0.1V$	relative humidity $65\% \pm 5\%$ $V_h: 5V \pm 0.1V$	$RL: 10K\Omega \pm 5\%$ $V_h: 1.4V \pm 0.1V$
	Preheat time	No less than 48 hours	Detecting range: 20ppm-2000ppm carbon monoxide	

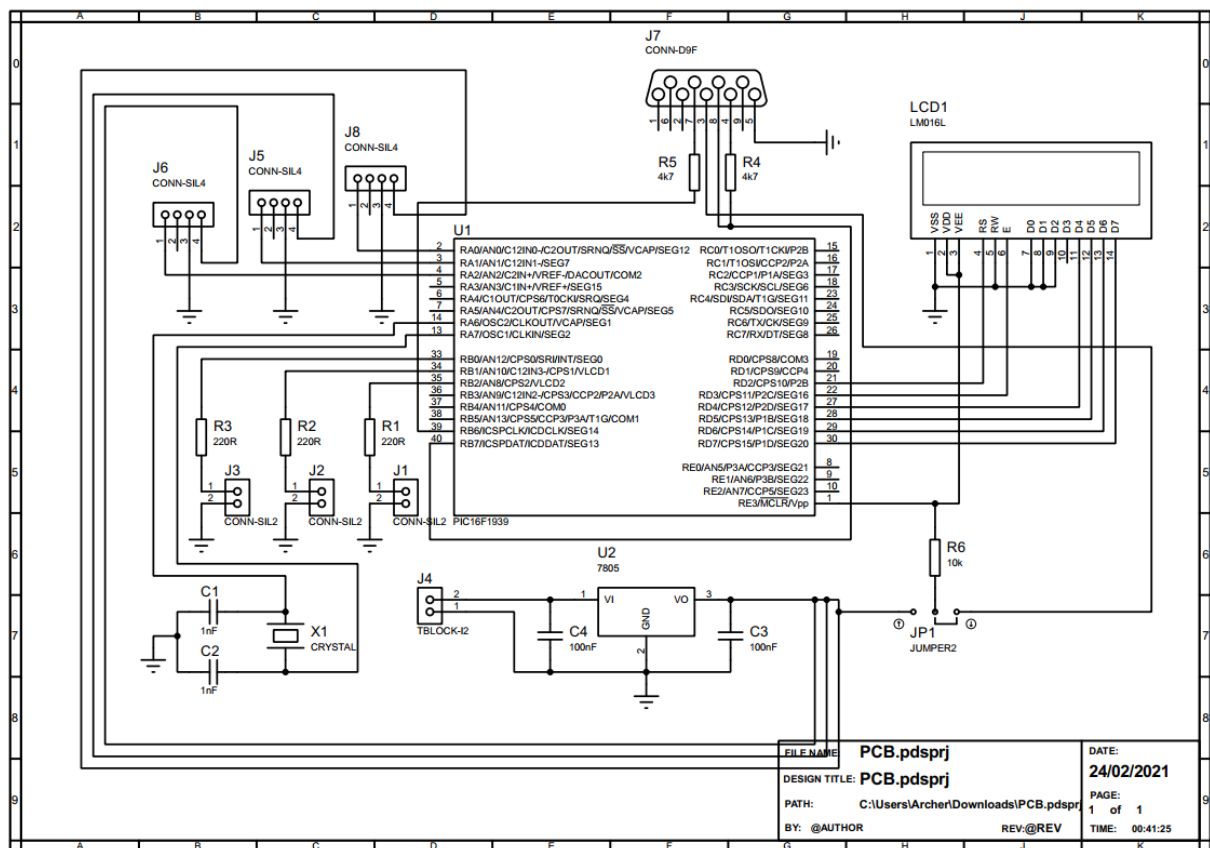
MQ-136	Symbol	Parameter name	Technical parameter	Remark 2
	$R_s$	Sensing Resistance	30K $\Omega$ -200K $\Omega$ (10ppm $H_2S$ )	Detecting concentration scope: 1-100ppm $H_2S$
	$\alpha$ (20/5) $H_2S$	Concentration Slope rate	$\leq 0.65$	
	Standard Detecting Condition	Temp: 20℃ $\pm$ 2℃      Vc:5V $\pm$ 0.1 Humidity: 65% $\pm$ 5%    Vh: 5V $\pm$ 0.1		
	Preheat time	Over 24 hour		

Slika 3: Karakteristika osjetljivosti

## 1.6 Shema u programskom okruženju Proteus

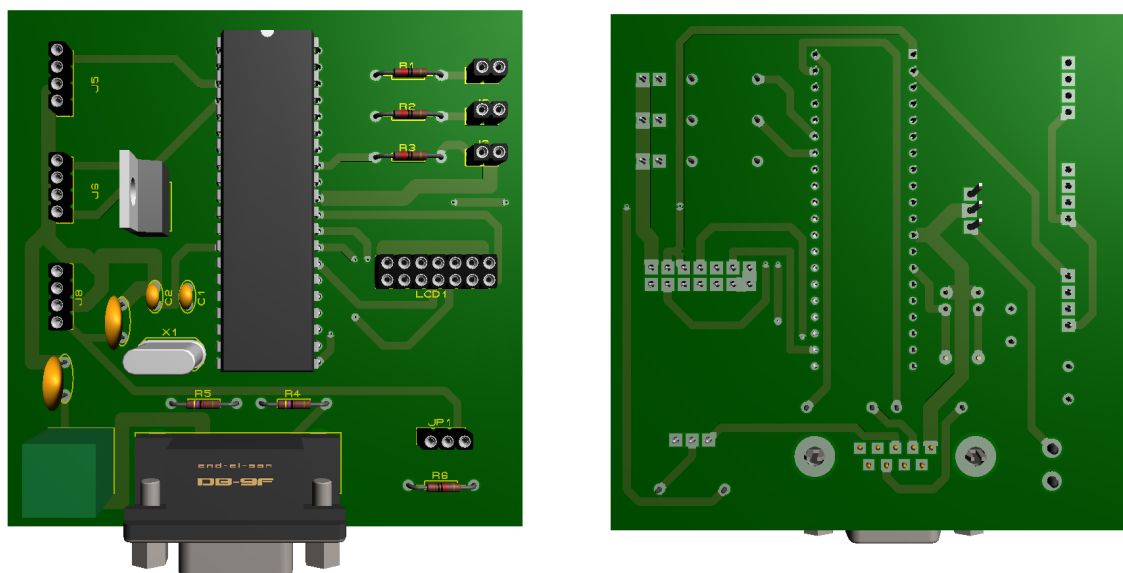
Konektori J1, J2, J3 predstavljaju mjesta na koje se ubace LED diode, odabran je takav dizajn, da ukoliko se pojavi potreba da se zamijeni neka od LED-ica, vrlo lagano se može izvršiti zamjena. Konektori J5, J6, J8 predstavljaju mjesta na koje se ubace senzori, isto u svrhu lakše zamjene pojedinih komponenti na uređaju. Na konektor J4 se dovodi napajanje iz naponske jedinice. Postavljen je i J7 konektor za programator čija je svrha mogućnost reprogramiranja uređaja. Većina elektronike je zamjenjiva vrlo lagano, što dovodi do velike fleksibilnosti pri primjeni ovakvog uređaja.





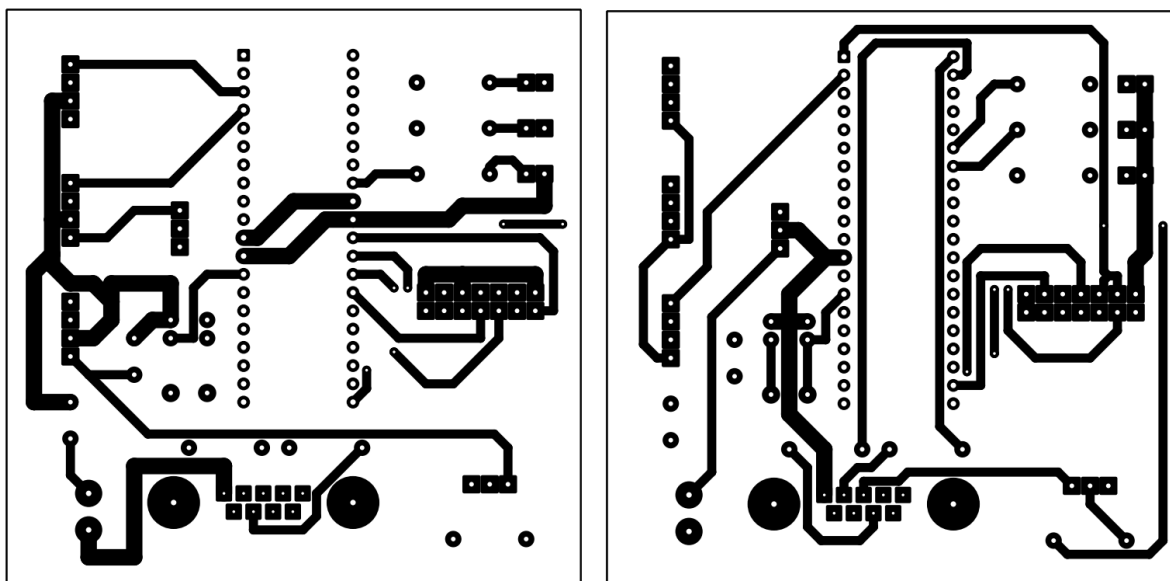
Slika 4: Shema projektnog zadatka

## 1.7 3D prikaz Sheme u programskom okruženju Proteus



Slika 5: 3D prikaz donje i gornje strane PCB ploče u Proteus-u

## 1.8 PCB

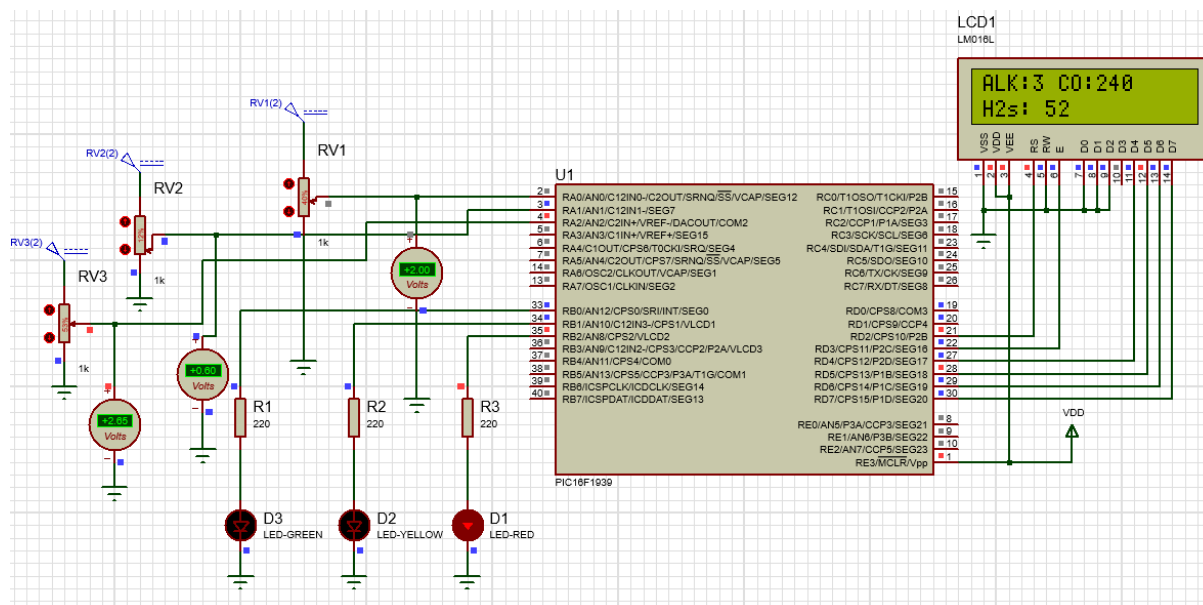


Slika 6: Prikaz donje i gornje strane PCB ploče sa akcentom na vodove u Proteus-u

## 1.9 Rezultati simulacije

### Napomena:

Korištena je ilustrativna šema u kojoj potencijometri mijenjaju funkciju senzora. Rezultati su dati u nastavku.



Kratak video testiranja funkcionalnosti možete pogledati na linku:

<https://drive.google.com/file/d/1jHdU0onYws9EC1d1vVOZHJJ1DbmzxUKS/view?usp=sharing>