



Ардуино радионица



# ARDUINO RADIONICA

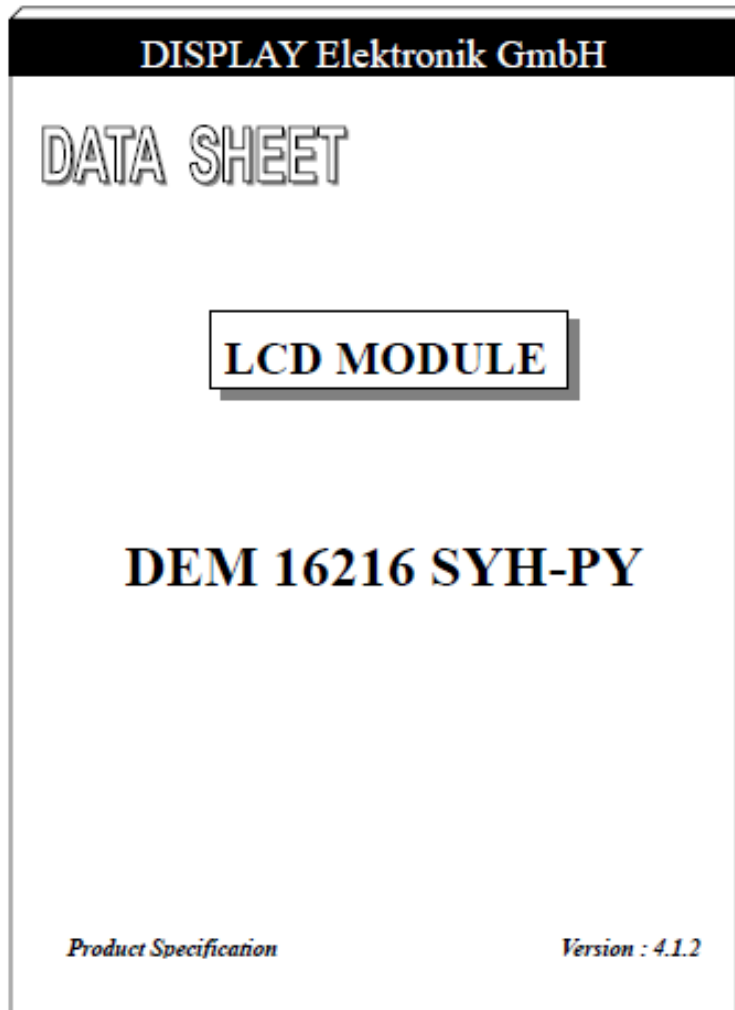


- Tradicionalna nauka razlikuje tri osnovna stanja u kojima može postojati materija: čvrsto, tečno i gasovito. Međutim, sredinom 19. veka, naučnici su primetili da se neke supstance ponašaju neobično. Bile su raznobojne i nisu se povinovale uobičajenim optičkim svojstvima bilo kog poznatog agregatnog stanja. Vremenom su dokazi naučnika, pogotovo Nemca Ota Lemana i austrijanca Fridriha Rajnicera, dovelil do toga da bude opšteprihvaćen novi oblik materije, poznat pod imenom **tečni kristal**.
- Tečnosti su izotropske i nemaju nikakav smer, dok kristali imaju jasno definisane vektore smeru koji opisuju karakteristike strukture rešetke. Tečni kristali mogu imati samo jedan ili dva smeru; dakle u stanju su da teku poput tečnosti, ali imaju i neke parametre slične materiji u čvrstom stanju.

- Najpoznatija primena tečnih kristala je **ekran od tečnih kristala** (liquid crystal display – LCD). LCD ekrane je šezdesetih godina 20. veka počeo da primenjuje američki naučnik Džejms Fergason. Oni su zasnovani na interakciji tečnog kristala sa polarizovanim svetlosnim talasima i električnim poljem. Oblik LCD-a koji je najčešće u upotrebi je numerički displej sa sedam segmenata i koristi se kod ručnih i drugih satova i sličnih predmeta.
- Osim LCD-a, postoji veliki broj drugih primena tečnih kristala, a još ih se mnogo razvija. Na primer, optička svojstva holesteričnog tečnog kristala čine ga veoma upotrebljivim kod termometara koji menjaju boju dok se menja temperatura. Takođe, postoje „**pametni prozori**“ koji koriste tečne kristale i struju da promene stanje prozora, od providnog do delimično providnog uz pomoć prekidača, što omogućava privatnost bez gubitka svetlosti koja ulazi kroz prozor.

- Displej ima 16 pinova od kojih 8 predstavlja 8-bitnu magistralu za podatke i tri kontrolna pina. Ostali pinovi koriste se za napajanje logike LCD-a i pozadinskog osvetljenja.





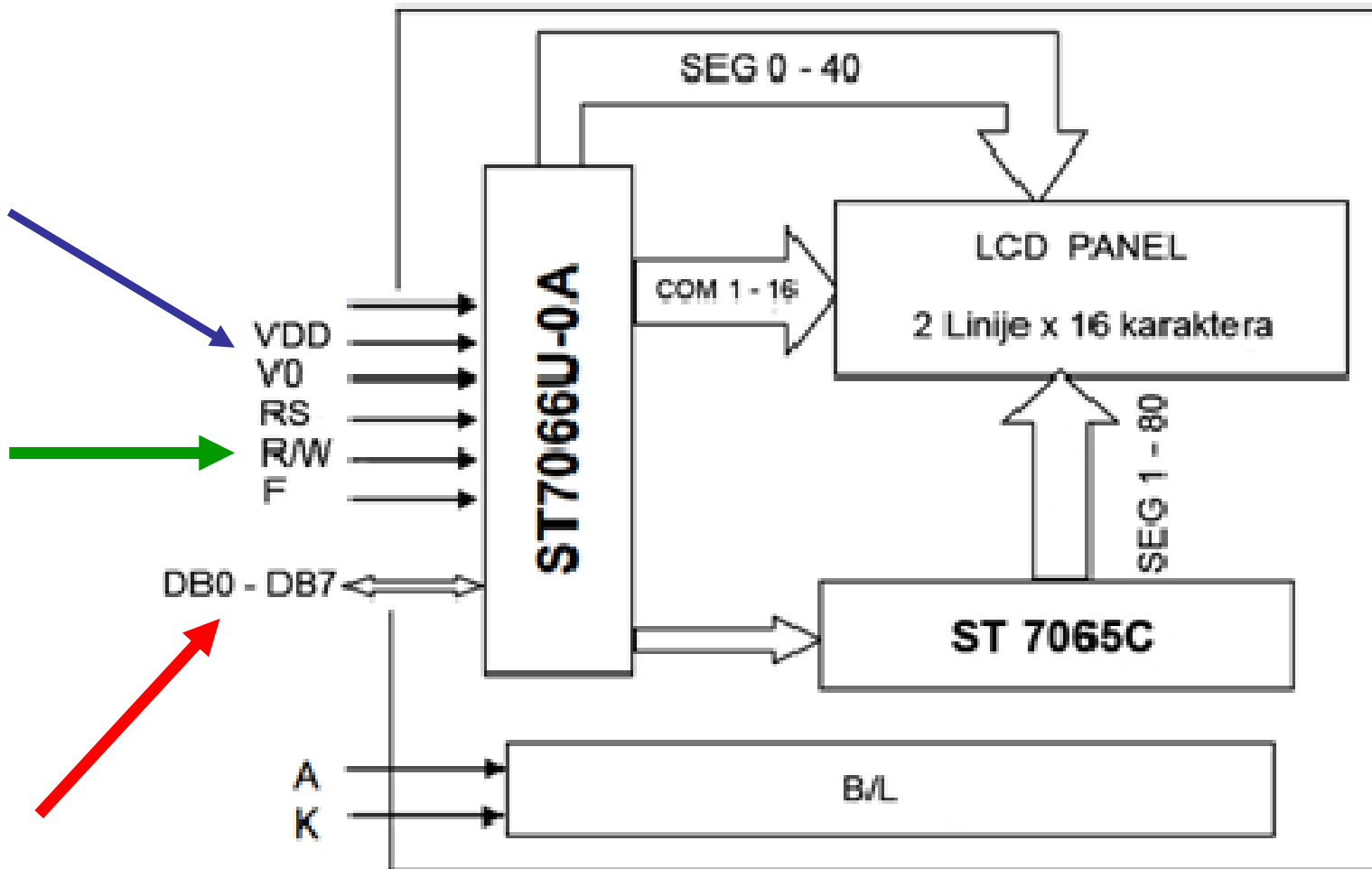
*DEM 16216 SYH-PY*

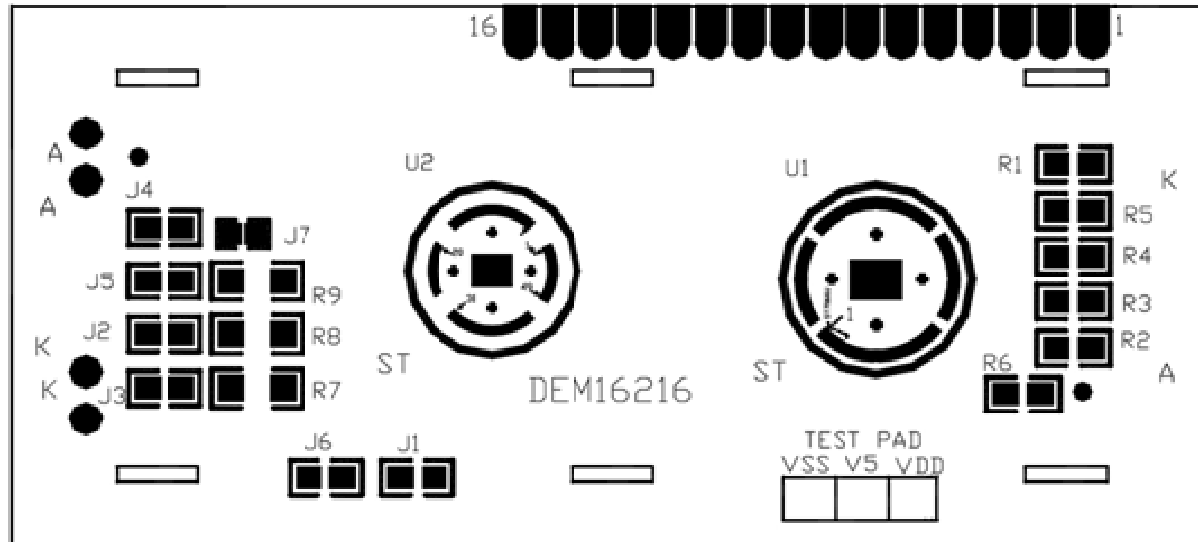
*Product Specification*

**CONTENTS**

1. FUNCTIONS & FEATURES .....	2
2. MECHANICAL SPECIFICATIONS .....	2
3. EXTERNAL DIMENSIONS .....	3
4. BLOCK DIAGRAM .....	4
5. PIN ASSIGNMENT .....	4
6.1 PCB DRAWING AND DESCRIPTION .....	5
6.2 EXAMPLE APPLICATION .....	6
7. BACKLIGHT ELECTRICAL AND OPTICAL SPECIFICATIONS .....	7
8. MAXIMUM ABSOLUTE POWER RATINGS .....	8
9. ELECTRICAL CHARACTERISTICS .....	8
10. DISPLAY DATA RAM (DDRAM) .....	11
11. INSTRUCTION DESCRIPTION .....	12
12. INITIALIZING BY INSTRUCTION .....	13
13. CHARACTER GENERATOR ROM (ST7066-0A) .....	15
14. LCD MODULES HANDLING PRECAUTIONS .....	16
15. OTHERS .....	16

Broj pina	Naziv pina	Nivo	Funkcija
1	V <sub>ss</sub>	0 V	Masa
2	V <sub>dd</sub>	+5 V	Napajanje (4,7 ÷ 5,3 V)
3	V <sub>o</sub>	0 - 5 V	Jačina kontrasta LCDa
4	RS	L/H	Selekcija registra, 1 = na DB0÷DB7 nalazi se podatak u ASCII formatu, 0 = na DB0÷DB7 nalazi se instrukcija
5	R/W	L/H	1 = čitanje podataka, 0 = upis podataka
6	E	H → L	E = enable, omogućava pristup LCD-u
7	DB0	L/H	Data bit0
8	DB1	L/H	Data bit1
9	DB2	L/H	Data bit2
10	DB3	L/H	Data bit3
11	DB4	L/H	Data bit4
12	DB5	L/H	Data bit5
13	DB6	L/H	Data bit6
14	DB7	L/H	Data bit7
15	LED - (K)	L/H	Napajanje za pozadinsko osvetljenje (-)
16	LED +(A)	L/H	Napajanje za pozadinsko osvetljenje (+)





- **LCD se obično konfigurira da radi u četvorobitnom modu, gde se za prenos instrukcije i podataka koriste četiri linije DB4÷DB7 (pinovi DB0÷DB3 su vezani na masu).**





*CG RAM (Character generator RAM) – sadrži matrične tačkaste prikaze karaktera koji se mogu prikazati na LCD-u*

NO.7066-0A

b7-b4 b3-b0	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)			0	1	2	3	4	5	6	7	8	9	A	B	C
0001	(2)	!	@	#	\$	%	&	'	(	)	*	+	=	-	.	/
0010	(3)	"	#	\$	%	&	'	(	)	*	+	=	-	.	/	:
0011	(4)	#	\$	%	&	'	(	)	*	+	=	-	.	/	:	;
0100	(5)	\$	%	&	'	(	)	*	+	=	-	.	/	:	;	<
0101	(6)	%	&	'	(	)	*	+	=	-	.	/	:	;	<	=
0110	(7)	&	'	(	)	*	+	=	-	.	/	:	;	<	=	>
0111	(8)	'	(	)	*	+	=	-	.	/	:	;	<	=	>	?
1000	(1)	(	)	*	+	=	-	.	/	:	;	<	=	>	?	!
1001	(2)	)	*	+	=	-	.	/	:	;	<	=	>	?	!	@
1010	(3)	*	+	=	-	.	/	:	;	<	=	>	?	!	@	#
1011	(4)	+	=	-	.	/	:	;	<	=	>	?	!	@	#	\$
1100	(5)	=	-	.	/	:	;	<	=	>	?	!	@	#	\$	%
1101	(6)	-	.	/	:	;	<	=	>	?	!	@	#	\$	%	&
1110	(7)	.	/	:	;	<	=	>	?	!	@	#	\$	%	&	'
1111	(8)	/	:	;	<	=	>	?	!	@	#	\$	%	&	'	(

- Ova biblioteka omogućava Arduino pločici da upravlja LCD-om koji je projektovan na bazi čipa **HD44780** (ili kompatibilnog) i koji se najčešće susreće kod tekstualnih LCD-ova. Biblioteka može da radi i u 4-bitnom i u 8-bitnom režimu (tj. može da koristi 4 ili 8 linija podataka pored **RS**, **Enable** i opcione **RW** kontrolne linije).
- Raspolaže sledećim funkcijama:

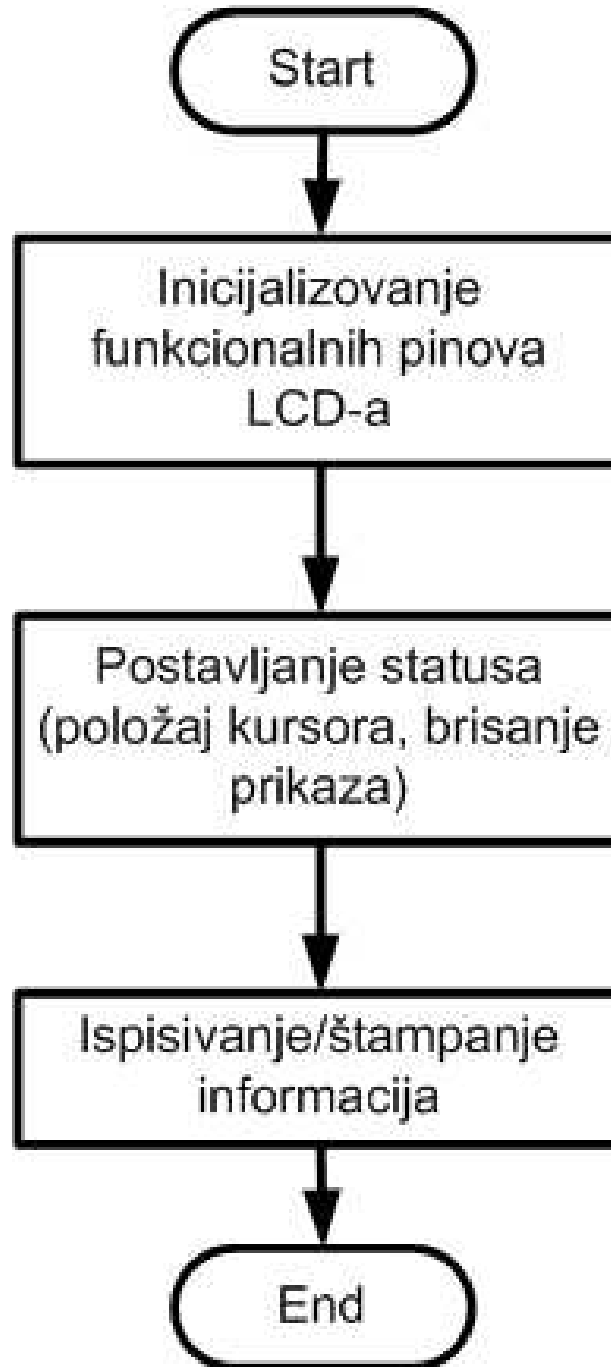
**LiquidCrystal()**  
**begin()**  
**clear()**  
home()  
**setCursor()**  
write()  
**print()**

cursor()  
noCursor()  
blink()  
noBlink()  
display()  
noDisplay()

**scrollDisplayLeft()**  
**scrollDisplayRight()**  
autoscroll()  
noAutoscroll()  
leftToRight()  
rightToLeft()  
createChar()

- **LiquidCrystal()** – kreira promenljivu tipa LiquidCrystal. Prikaz se može kontrolisati koristeći 4 ili 8 linija podataka. Ako se koristi 4-bitni režim, tada se linije d0 do d3 ostavljaju nepovezane a **RW** pin se može priključiti na GND (zemlju) umesto na Arduino pin.
- Sintaksa:
  - LiquidCrystal(rs, enable, d4, d5, d6, d7)
  - LiquidCrystal(rs, rw, enable, d4, d5, d6, d7)
  - LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7)
  - LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7)

- **Parametri:**
- **rs:** broj Arduino pina koji se povezuje na RS pin LCD-a
- **rw:** broj Arduino pina koji se povezuje na RW pin LCD-a (*opciono*)
- **enable:** broj Arduino pina koji se povezuje na enable pin LCD-a
- **d0, d1, d2, d3, d4, d5, d6, d7:** brojevi Arduino pinova koji se povezuju na odgovarajuće pinove podataka LCD-a. d0, d1, d2 i d3 su opciono; ako se ne koriste, LCD će biti kontrolisan koristeći samo 4 linije podataka (d4, d5, d6, d7).



```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);
```

```
void setup()
```

```
{
```

```
    lcd.begin(16,1);
```

```
    lcd.print("hello, world!");
```

```
}
```

```
void loop() {}
```

- U ovom slučaju linija, odnosno pin,
- broj 5 (**R/W**) vezana je na Arduino pin 11 za čitanje podataka sa LCD-a.
- Od kontrolnih linija koristimo još, **E** koja se povezuje sa Arduino pinom 10 i
- **RS** koja se povezuje sa Arduino pinom 12.
- **Napomena:** LCD pin 15 (K) se može povezati sa Arduino pinom 13 za kontrolu pozadinskog osvetljenja.

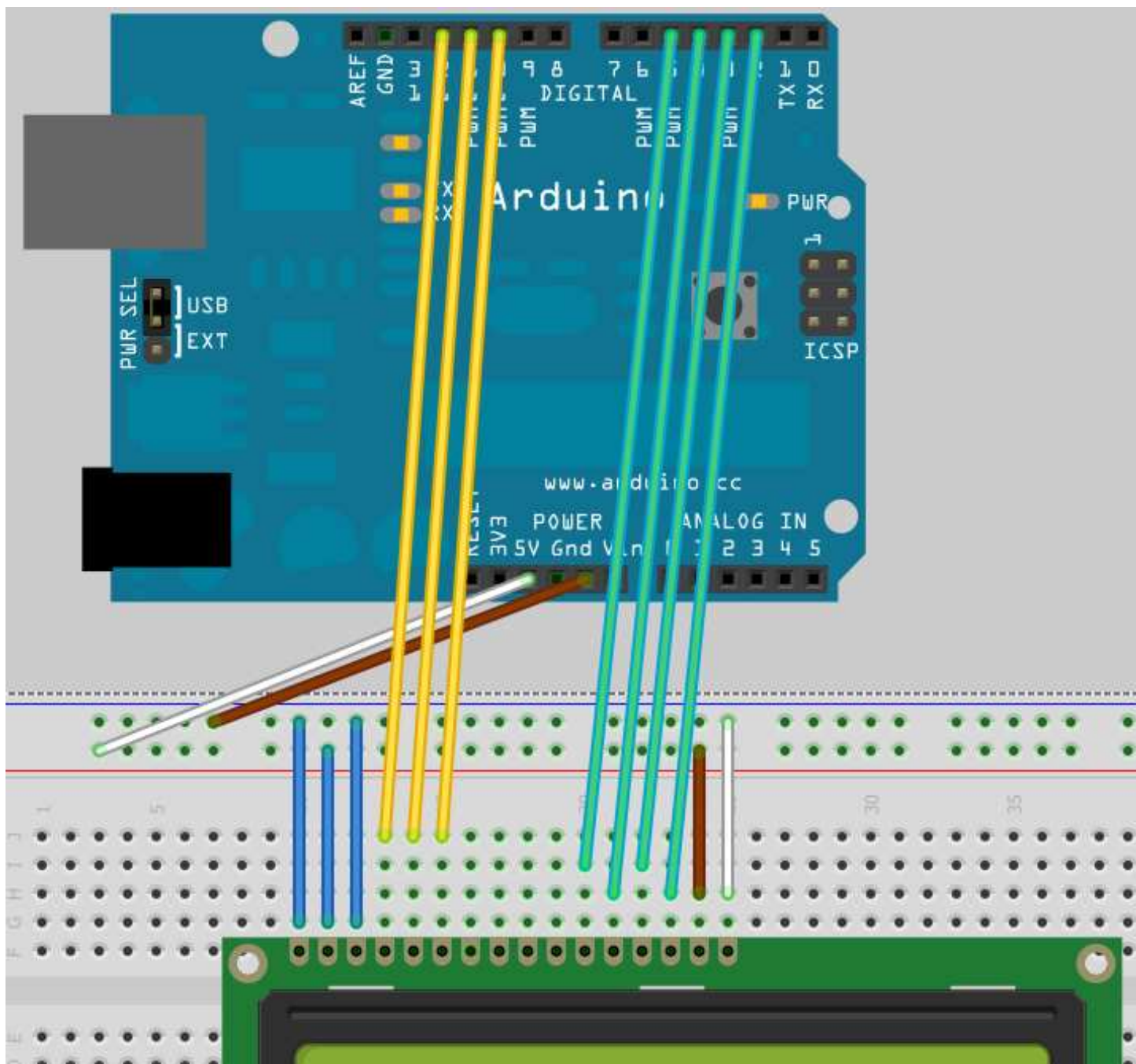
LCD Pin	Priključen na:
1 (VSS)	GND Arduino pin*
2 (VDD)	+ 5v Arduino pin
3 (kontrast)	Otpornik ili potencijometar na GND Arduino pin**
4 RS	Arduino pin 12
5 R/W	Arduino pin 11
6 Enable	Arduino pin 10
7 Nema konekcije	
8 Nema konekcije	
9 Nema konekcije	
10 Nema konekcije	
11 (Data 4)	Arduino pin 5
12 (Data 5)	Arduino pin 4
13 (Data 6)	Arduino pin 3
14 (Data 7)	Arduino pin 2
15 Backlight +	Otpornik na Arduino pin 13***
16 Backlight GND	GND Arduino pin*

\* Koristiti prototipsku ploču za realizaciju GND čvorišta

\*\* Ukoliko se koristi potencijometar, srednji izvod se povezuje sa LCD pinom 3 a jedan od krajeva sa GND

\*\*\* Preporučuje se korišćenje otpornika bar od 40 oma za ograničenje struje

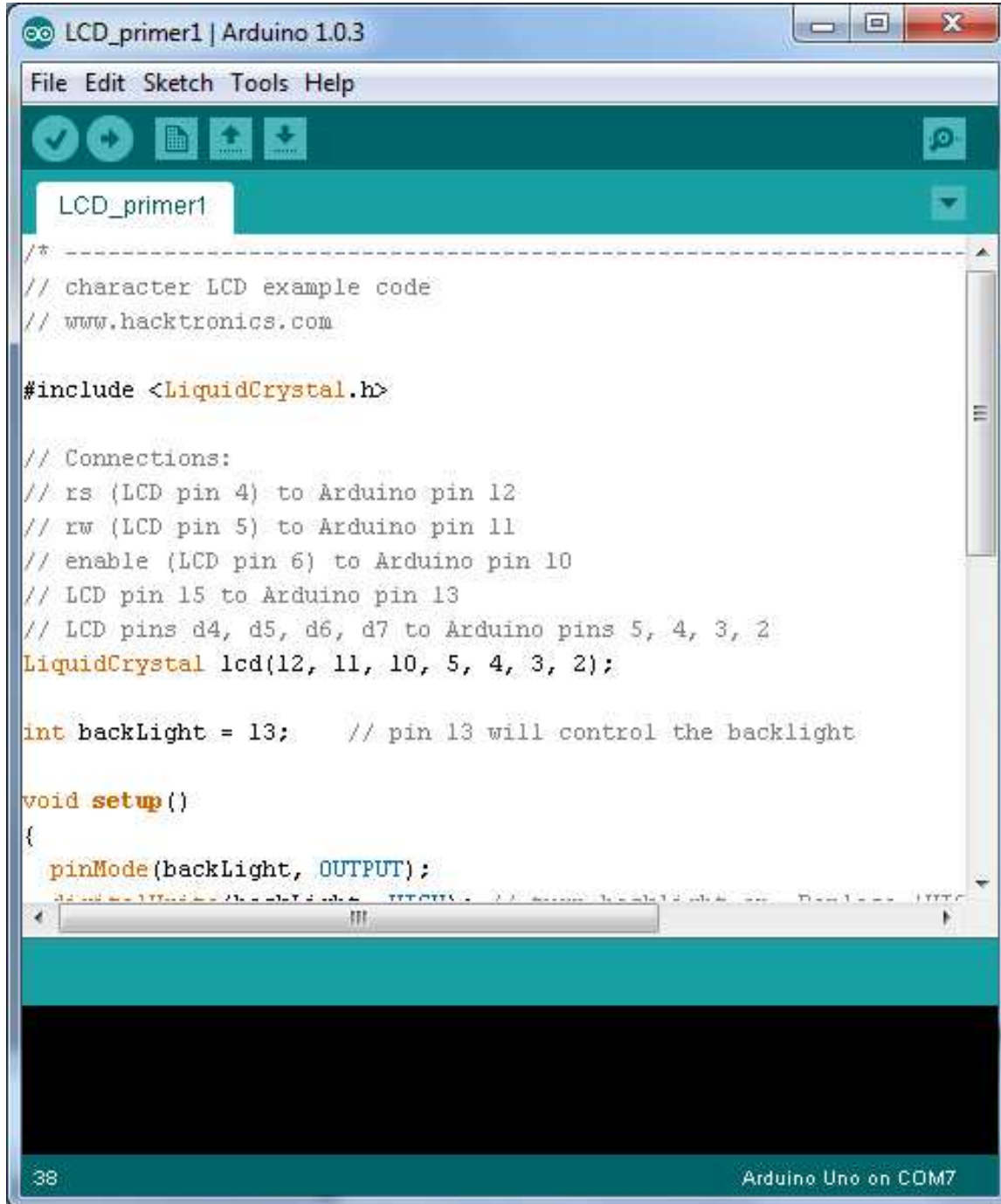




```
/* ----- */  
// character LCD example code  
#include <LiquidCrystal.h>  
  
// Connections:  
// rs (LCD pin 4) to Arduino pin 12  
// rw (LCD pin 5) to Arduino pin 11  
// enable (LCD pin 6) to Arduino pin 10  
// LCD pin 15 to Arduino pin 13  
// LCD pins d4, d5, d6, d7 to Arduino pins 5, 4, 3, 2  
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);  
  
int backLight = 13; // pin 13 will control the backlight
```

```
void setup()
{
  pinMode(backLight, OUTPUT);
  digitalWrite(backLight, HIGH); // turn backlight on. Replace 'HIGH' with
                                // 'LOW' to turn it off.
  lcd.begin(16,2);              // columns, rows. use 16,2 for a 16x2 LCD, etc.
  lcd.clear();                  // start with a blank screen
  lcd.setCursor(0,0);           // set cursor to column 0, row 0 (the first row)
  lcd.print("*ARDU RADIONICA*"); // change this text to whatever you
                                // like. keep it clean.
  lcd.setCursor(0,1);           // set cursor to column 0, row 1
  lcd.print("Zdravo svima!");
}

void loop()
{
}
/* ----- */
```



```
LCD_primer1 | Arduino 1.0.3
File Edit Sketch Tools Help
LCD_primer1
/* -----
// character LCD example code
// www.hacktronics.com

#include <LiquidCrystal.h>

// Connections:
// rs (LCD pin 4) to Arduino pin 12
// rw (LCD pin 5) to Arduino pin 11
// enable (LCD pin 6) to Arduino pin 10
// LCD pin 15 to Arduino pin 13
// LCD pins d4, d5, d6, d7 to Arduino pins 5, 4, 3, 2
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

int backlight = 13; // pin 13 will control the backlight

void setup()
{
  pinMode(backlight, OUTPUT);
  digitalWrite(backlight, HIGH); // turn backlight on. Defaults LOW!
  pinMode(13, OUTPUT);
}
```

38 Arduino Uno on COM7

```
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);

void setup() {
// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("Zdravo svima!");
delay(1000);
}
```

```
void loop() {
```

```
  // scroll 13 positions (string length) to the left
```

```
  // to move it offscreen left:
```

```
  for (int positionCounter = 0; positionCounter < 13; positionCounter++) {
```

```
    // scroll one position left:
```

```
    lcd.scrollDisplayLeft();
```

```
    // wait a bit:
```

```
    delay(350);
```

```
  }
```

**// scroll 29 positions (string length + display length) to the right**

**// to move it offscreen right:**

```
for (int positionCounter = 0; positionCounter < 29; positionCounter++) {
```

```
  // scroll one position right:
```

```
  lcd.scrollDisplayRight();
```

```
  // wait a bit:
```

```
  delay(350);
```

```
}
```

```
// scroll 16 positions (display length + string length) to the left
```

```
// to move it back to center:
```

```
for (int positionCounter =0; positionCounter < 16; positionCounter++) {
```

```
// scroll one position left:
```

```
lcd.scrollDisplayLeft();
```

```
// wait a bit:
```

```
delay(350);
```

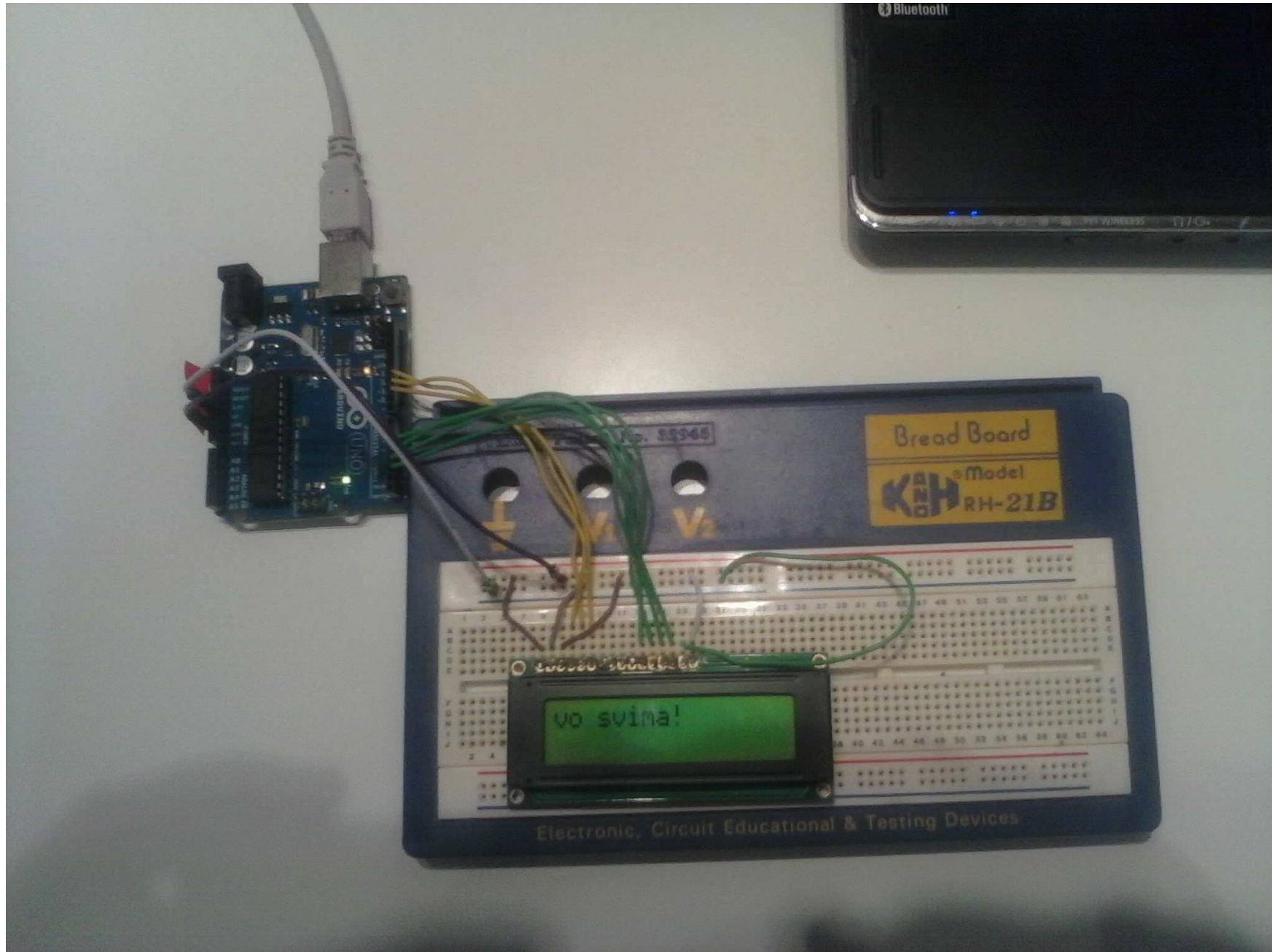
```
}
```

```
// delay at the end of the full loop:
```

```
delay(1000);
```

```
}
```





# Kreiranje svojih znakova

- Funkcija **createChar()** – kreira namenski karakter (glif) za prikazivanje na LCD-u. **Do osam karaktera dimenzija 5x8 piksela je podržano (numerisano od 0 do 7).** Pojavni oblik svakog namenskog karaktera se specificira kao niz od 8 bajtova, za svaki red po jedan.
- Pet najmanje značajnih bitova svakog bajta definišu piksele tog reda.
- Napomena : Kada se referencira namenski karakter "0", ako nije promenljiva, potrebno je da se referencira kao bajt, u protivnom će kompajler prijaviti grešku.

- Sintaksa:

***lcd.createChar(num, data)***

- Parametri:

**lcd:** promenljiva tipa LiquidCrystal

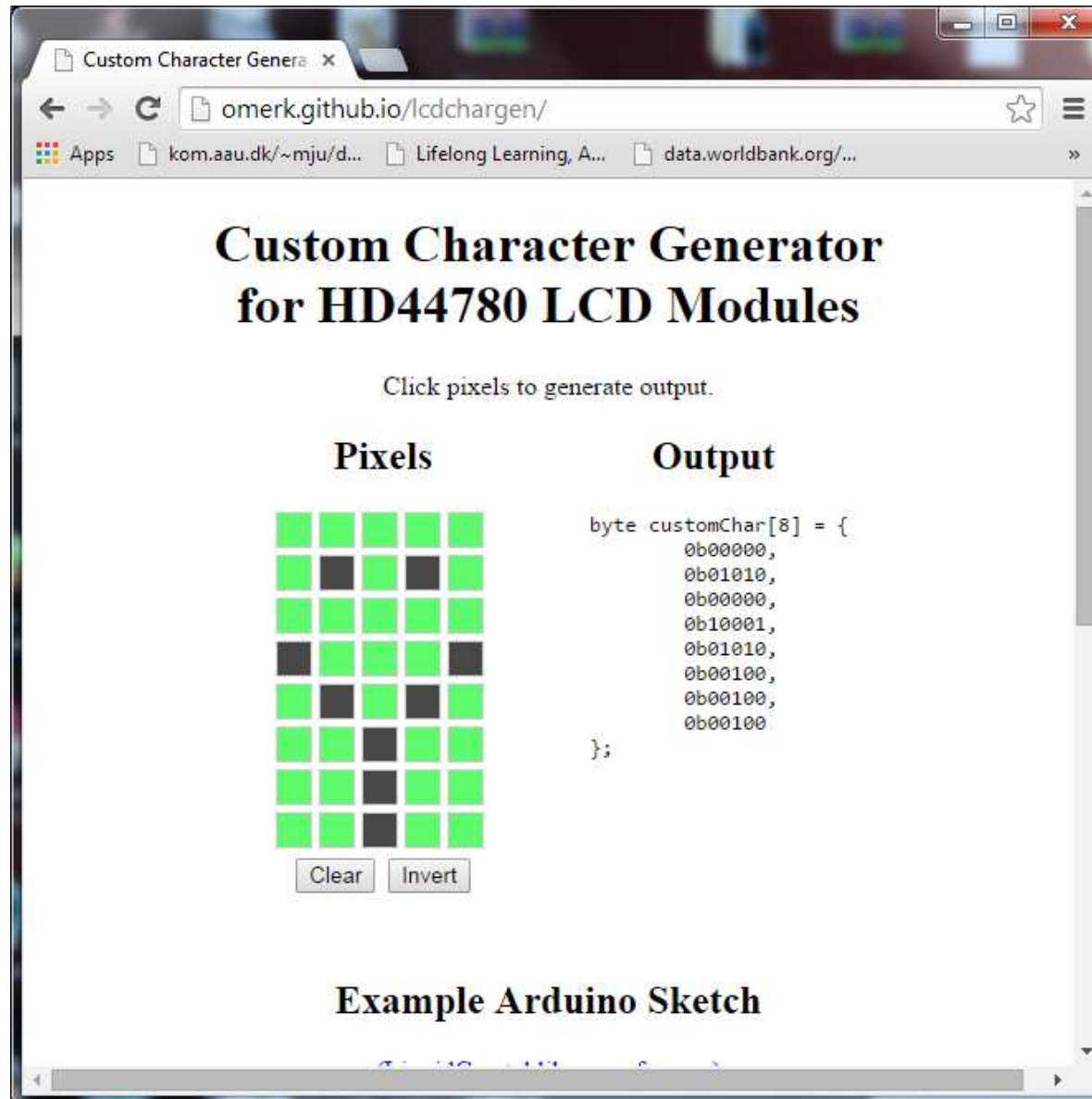
**num:** koji karakter se kreira (0 do 7)

**data:** pikselni podaci karaktera

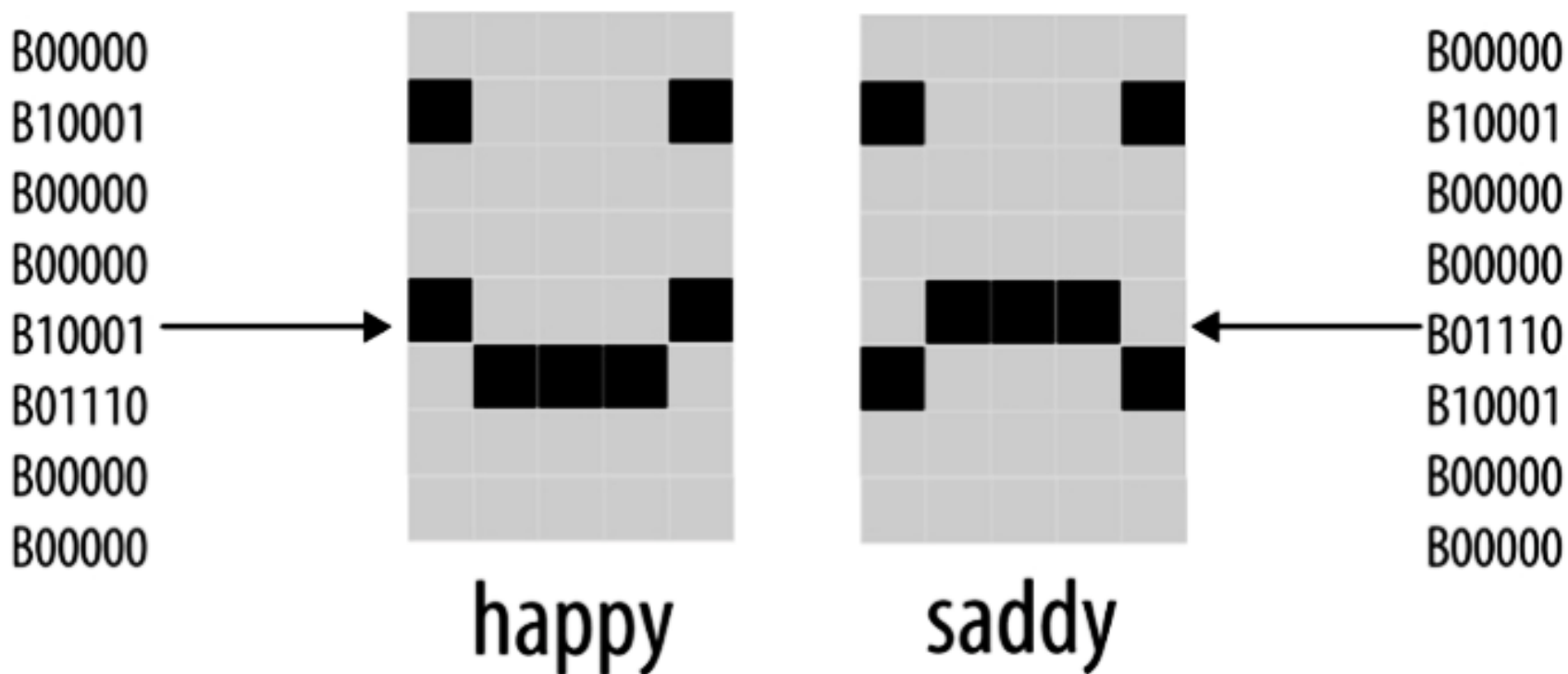
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
byte smiley[8] = {
    B00000,
    B10001,
    B00000,
    B00000,
    B10001,
    B01110,
    B00000,
};
void setup() {
    lcd.createChar(0, smiley);
    lcd.begin(16, 2);
    lcd.write(byte(0)); }
void loop() {}
```

- Alati na webu



- Definisanje namenskih znakova

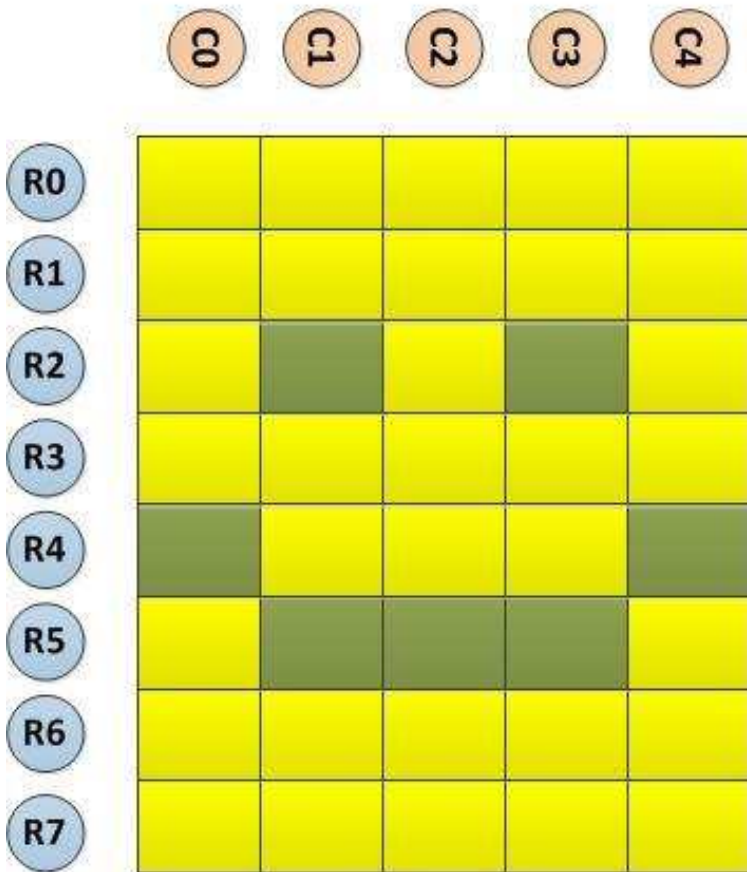


```
/* custom_char sketch creates an animated face using custom
   characters */
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);
byte happy[8] = { B00000, B10001, B00000, B00000, B10001, B01110,
                  B00000, B00000 };
byte saddy[8] = { B00000, B10001, B00000, B00000, B01110, B10001,
                  B00000, B00000 };
void setup() {
  lcd.createChar(0, happy);
  lcd.createChar(1, saddy);
  lcd.begin(16, 2); }

void loop() {
  for (int i=0; i<2; i++) {
    lcd.setCursor(0,0);
    lcd.write(i); delay(500); } }
```



- Niz 8\*5 piksela i odgovarajući binarni niz za **smile** simbol



**PIXEL ARRAY**

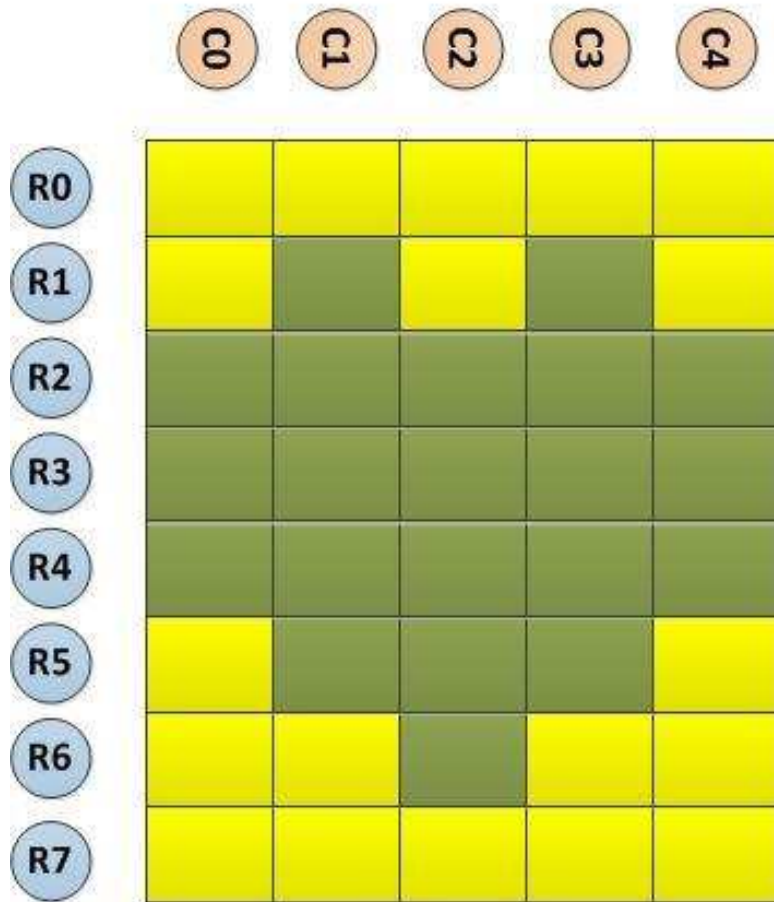
0	0	0	0	0
0	0	0	0	0
0	1	0	1	0
0	0	0	0	0
1	0	0	0	1
0	1	1	1	0
0	0	0	0	0
0	0	0	0	0

**BINARY ARRAY**

```
{
0b00000,
0b00000,
0b01010,
0b00000,
0b10001,
0b01110,
0b00000,
0b00000
};
```



- Niz 8\*5 piksela i odgovarajući binarni niz za simbol **srca**

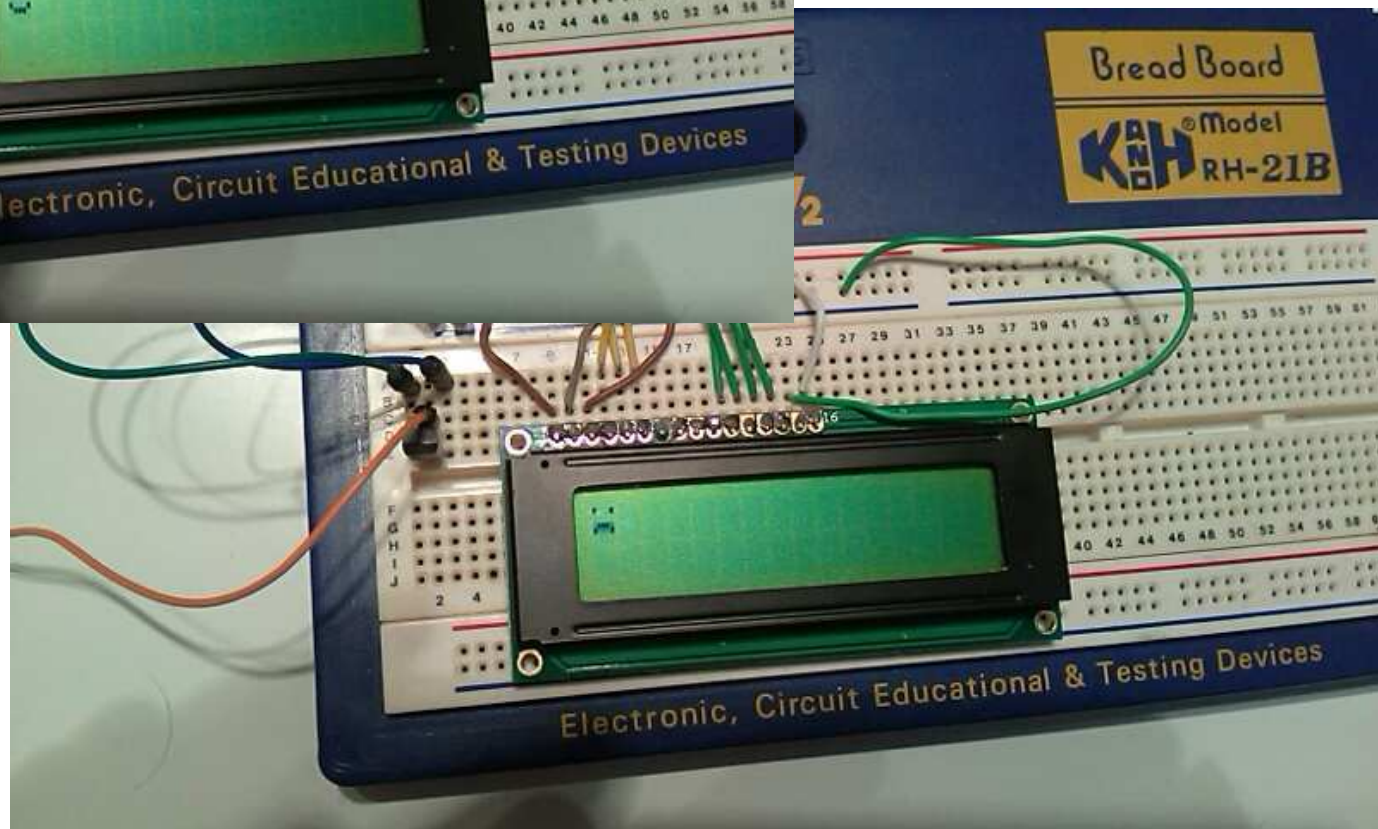
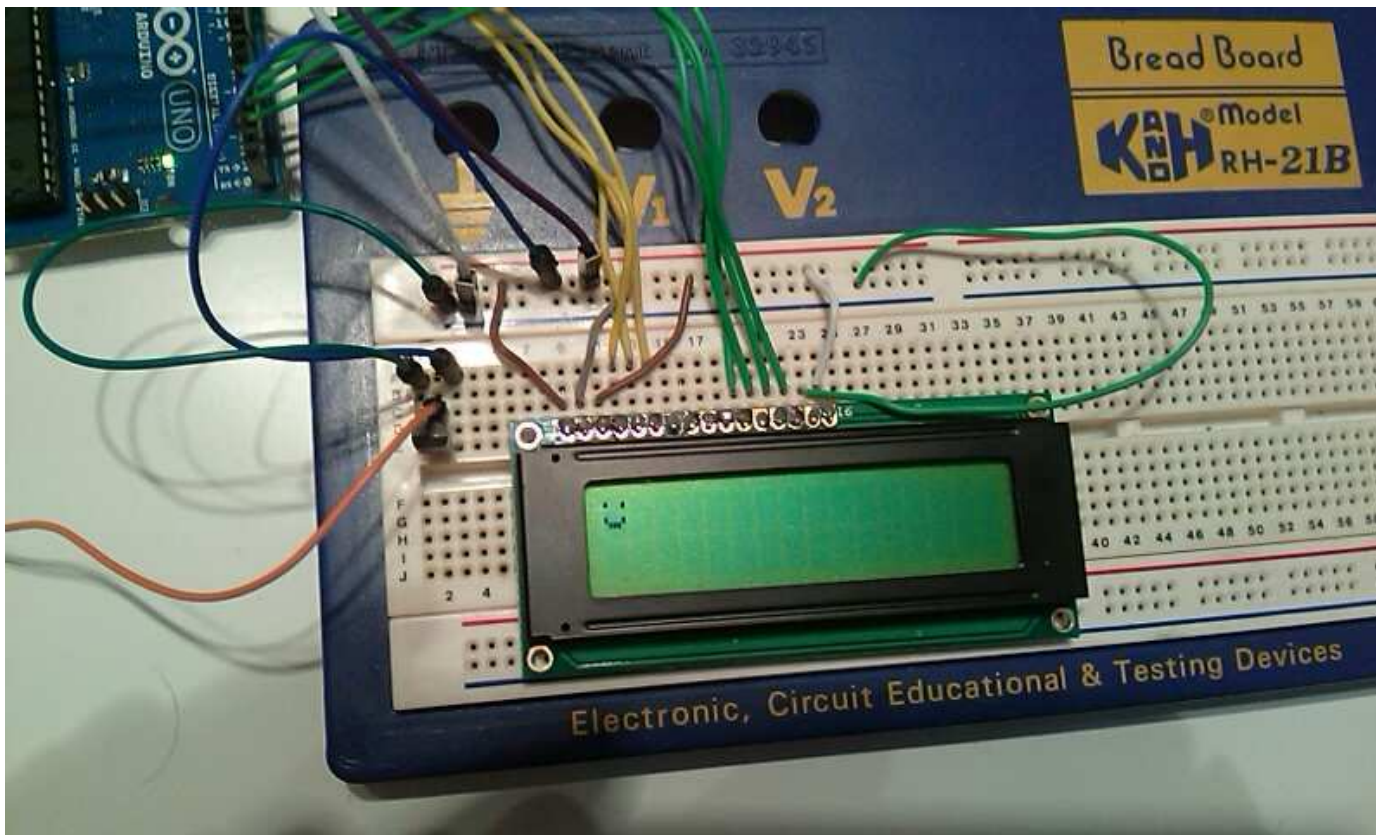


**PIXEL ARRAY**

0	0	0	0	0
0	1	0	1	0
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0
0	0	0	0	0

**BINARY ARRAY**

```
{
0b00000,
0b01010,
0b11111,
0b11111,
0b11111,
0b11111,
0b01110,
0b00100,
0b00000
};
```



- U ovom eksperimentu realizuje se jednostavni digitalni termometar koji koristi integrisani temperaturni senzor LM35DZ sa tzv. naponskim izlazom firme *National Semiconductors* i Arduino pločicu a izmerena temperatura se prikazuje u serijskom monitoru na personalnom računaru.
- Prema kataloškim podacima, poseduje osetljivost od 10 mV/°C a njegova funkcija prenosa je veoma linearna i odstupanje se nalazi u opsegu od  $\pm 0.1$  °C.



- Izlazni napon LM35DZ se može prikazati u obliku:

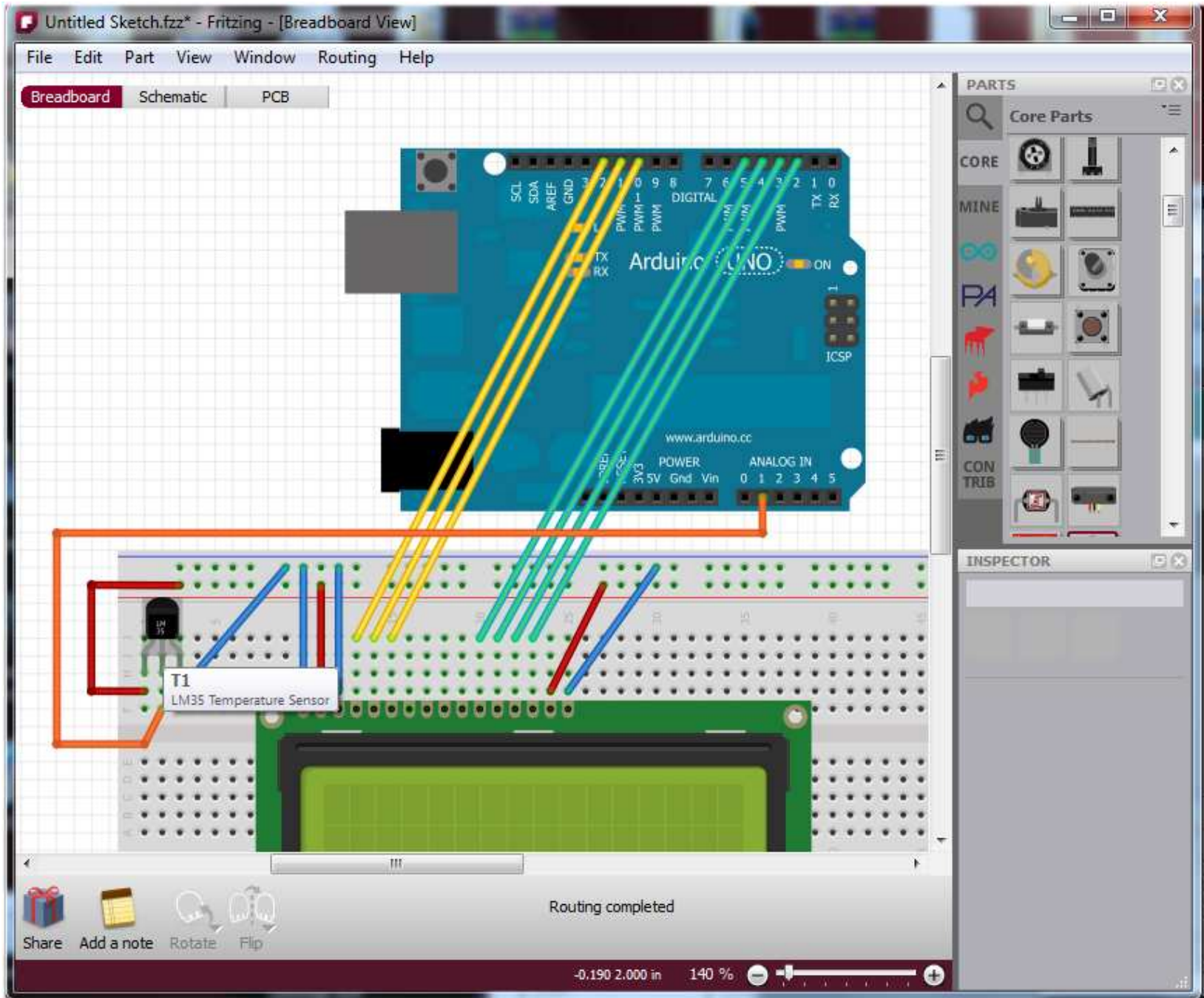
$$V_{out} = V_0 + aT,$$

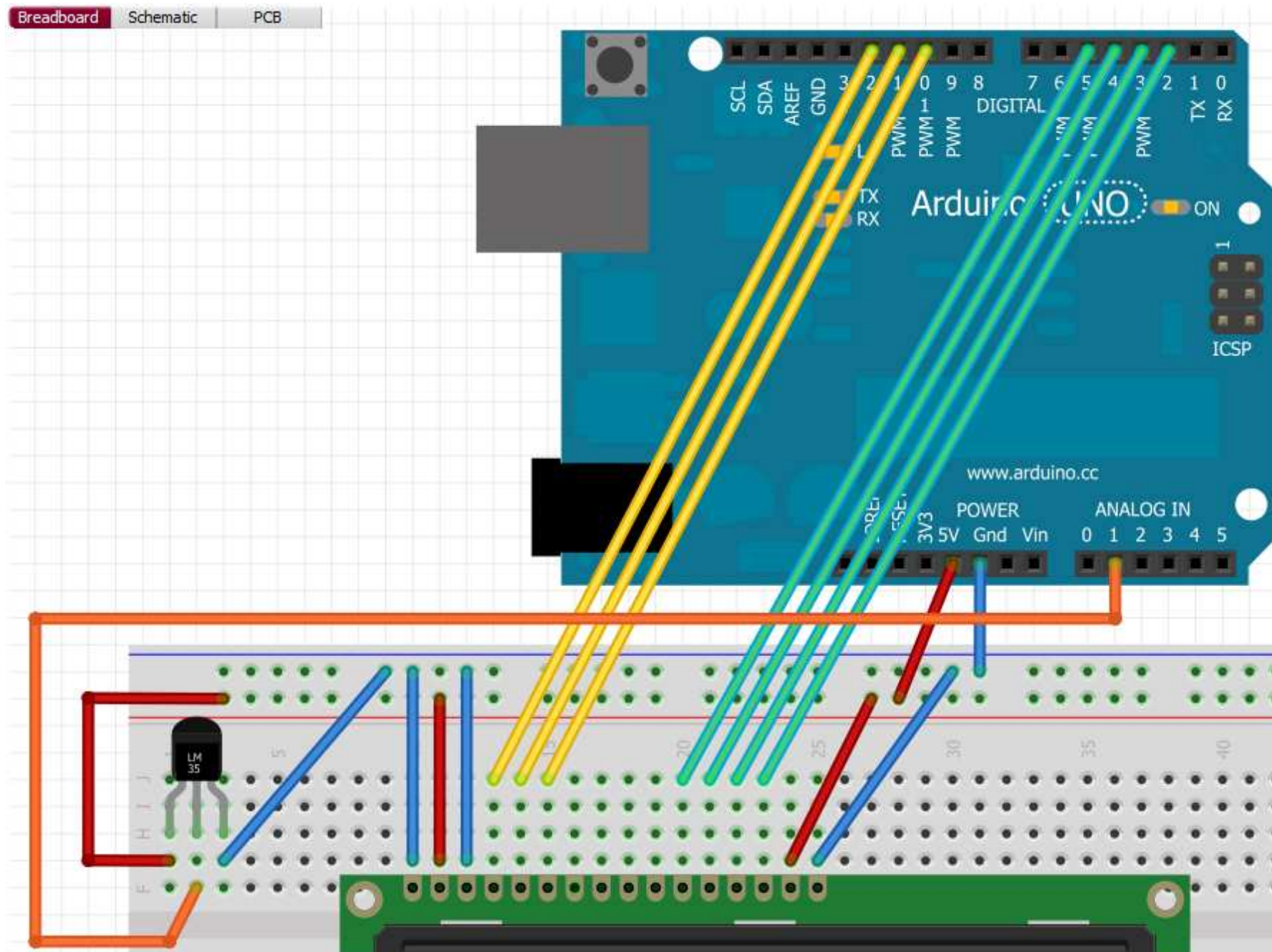
gde je  $T$  temperatura u K.  $V_0$  bi trebalo da 0, ali usled proizvodne varijacije može biti čak i 10 mV, što **može uneti grešku od 1 °C**, o čemu se mora voditi računa pri preciznijim merenjima. Takođe, osetljivost usled proizvodne tolerancije se kreće od 9.9 mV/°C do 10 mV/°C.

- Napon proporcionalan merenoj temperaturi dovodi se na pin A1. **Referentni napon** u ovom primeru je napon napajanja mikrokontrolera → **5 volti**.
- Mikrokontroler ima 10-bitni analogno/digitalni konvertor, tj. **daje vrednosti između 0 i 1023 za ulazni napon od 0 do 5 volti**.
- Senzor meri od 0 do 100 °C i na izlazu daje napon od 0 do 5 volti (5000 milivolti) → **1 stepenu odgovara 10 milivolti**
- Izmerena temperatura dobija se po formuli:

$$\text{((Vrednost sa A1) * 500) / 1023}$$









```
// Include the library to use a LCD display
#include <LiquidCrystal.h>
#define sensor 1 // Define the A1 pin as “sensor”

int Vin;          // Variable to read the value from the Arduino’s pin
float Temperature; // Variable that receives the converted voltage value
                  // to temperature

LiquidCrystal lcd (12, 11, 10, 5, 4, 3, 2);
/* The function above declares which Arduino’s pins
will be used for controlling the LCD */

void setup() {

    lcd.begin(16, 2); // It tells the Arduino that the display is a 16x2 type
    lcd.print("Temperatura: "); // Send the text to the screen of the display.
}
```

```
void loop() {
```

```
Vin = analogRead (sensor); // Tells the Arduino to read the pin and  
                             // stores the value in "Vin"
```

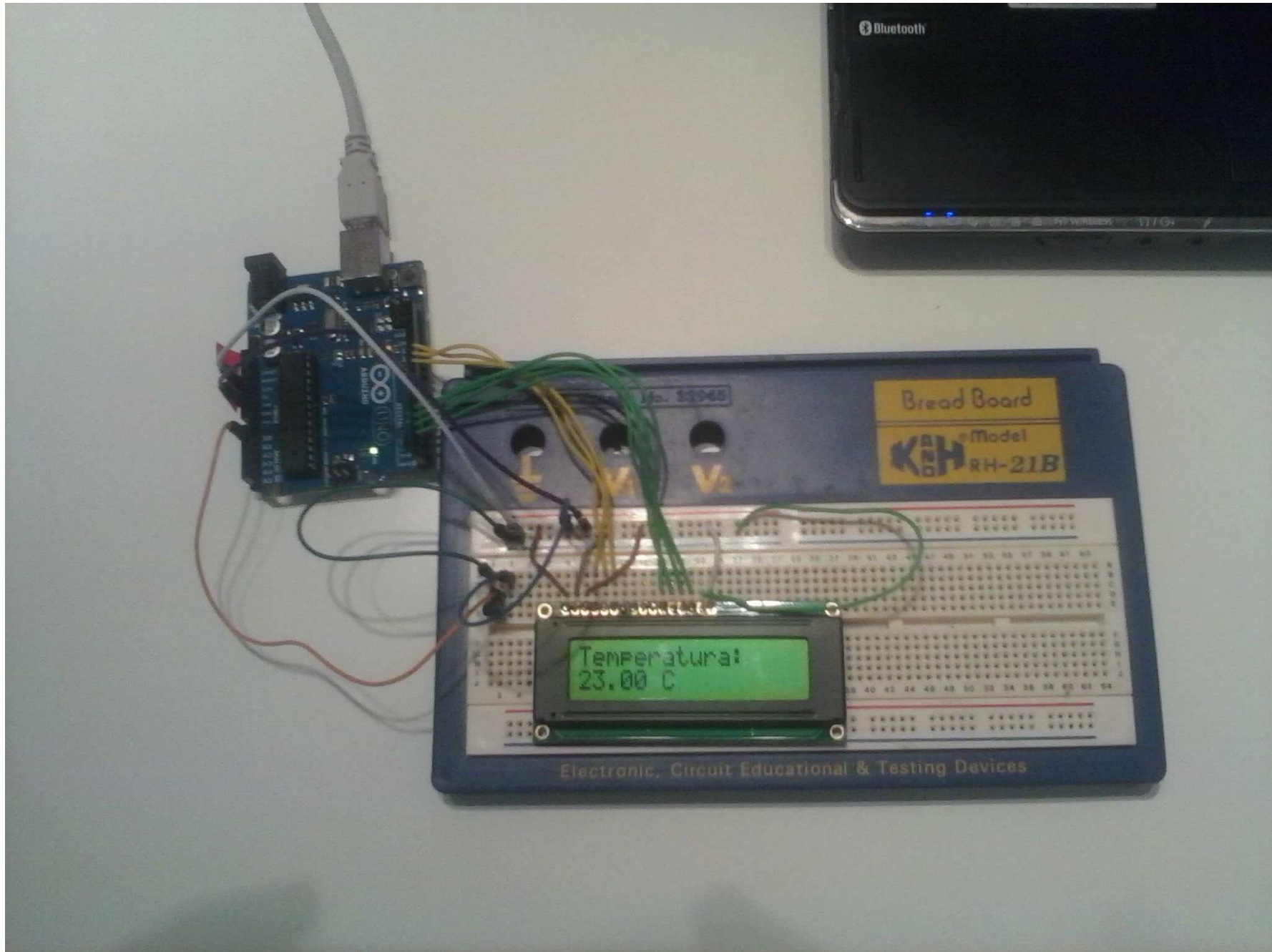
```
Temperature=(500*Vin)/1023; // Converts the voltage value into  
// temperature and stores it into the "Temperature" variable (in °C)
```

```
Icd.setCursor(0, 1); // Moves the cursor of the display to the next line  
Icd.print(Temperature); // Exhibits the value of the temperature on the  
                        // display
```

```
Icd.print(" C"); // Writes "C" to indicate that it is in Celsius scale.
```

```
delay(1000); // Waits for a second to read the pin again
```

```
}
```



### Ova prezentacija je nekomercijalna.

Slajdovi mogu da sadrže materijale preuzete sa Interneta, stručne i naučne građe, koji su zaštićeni Zakonom o autorskim i srodnim pravima. Ova prezentacija se može koristiti samo privremeno tokom usmenog izlaganja nastavnika u cilju informisanja i upućivanja studenata na dalji stručni, istraživački i naučni rad i u druge svrhe se ne sme koristiti –

**Član 44** - Dozvoljeno je bez dozvole autora i bez plaćanja autorske naknade za nekomercijalne svrhe nastave: (1) javno izvođenje ili predstavljanje objavljenih dela u obliku neposrednog poučavanja na nastavi; - ZAKON O AUTORSKOM I SRODNIM PRAVIMA ("Sl. glasnik RS", br. 104/2009 i 99/2011)

Dragan S. Marković