

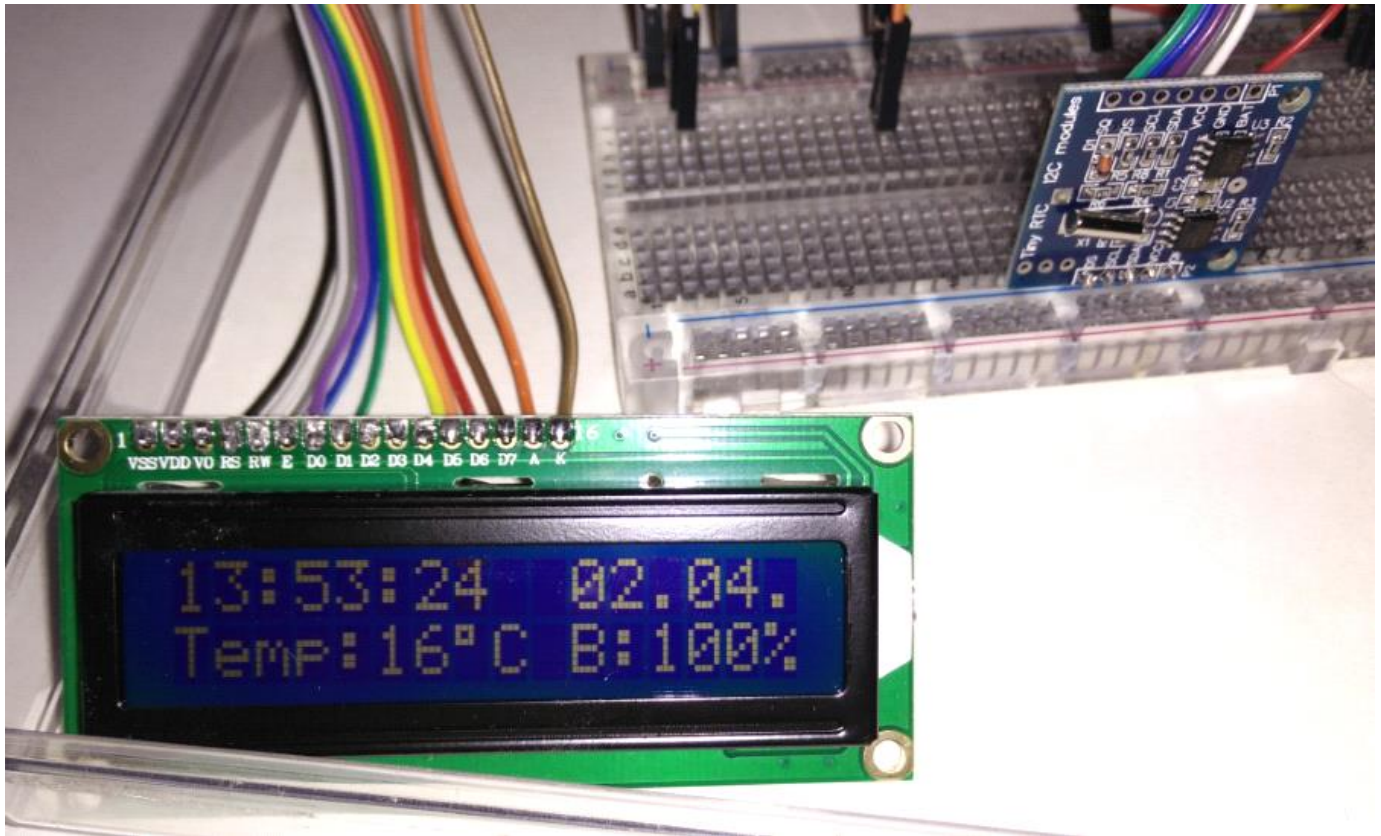
Arduino LCD sat

Autor: Darko Golner, e-mail: dgolner@inet.hr

Mali uvod

Za kućni demo projekt što bi mogao napraviti s Arduinoom zamislio sam LCD sat s pokazivačem temperature prostorije. Ništa preoriginalno i prekomplicirano, ali dovoljno petljivo što se tiče povezivanja žicama na malom prostoru Arduino pločice i breadboarda.

U gotovo svim Arduino starter kitovima na eBayu naišao sam na LCD modul i neki od senzora temperature. Kako ih pametno iskoristiti može se pronaći u dosta tutoriala na netu jer primijetio sam da demo sketchevi koji dolaze u starter kitovima nisu od neke velike koristi. Zbog toga sam razmislio ne bi li se tako često korišten LCD mogao iskoristiti za prikazivanje točnog vremena, datuma i temperature. Da bi sat bio precizan, potreban mi je i RTC (Real Time Clock) koji se može nabaviti za sitno dolara ili već dolazi u nekoj od verzija alternativnih Arduino starter kitova. Baš takav starter kit sam nabavio, a sadrži sve meni potrebne komponente. Iz njih sam odlučio izraditi LCD sat koji će biti napajan običnom 9 V baterijom, a prikazivat će datum, vrijeme, temperaturu i postotak baterije. Uz dosta žica, malo lemljenja i malo programiranja može se puno naučiti o tim komponentama koje se dalje mogu lakše koristiti i za druge projekte. Ovaj će projekt u konačnici izgledati ovako:



Što je sve potrebno?

Za ovaj projekt iskoristio sam jedan od Arduino UNO R3 starter kitova koji osim Arduino UNO R3 pločice sadrži i LCD te RTC modul. Pojedine komponente mogu se i samostalno nabaviti tako da nije nužnost nabava cijelog kita.

Ono što ću koristiti od komponenti u ovom tutorialu je nabrojano u nastavku:

- Arduino UNO R3 pločica ili neka druga kompatibilna
- breadboard pune veličine (big breadboard - 830 pinova)
- 1602 LCD
- RTC modul - Tiny RTC DS1307 ili neki vrlo sličan modul
- LM35 senzor temperature
- dva potenciometra - koristim jedan od 1 k Ω i jedan od 10 k Ω , ali mogu se koristiti i dva istog otpora
- držač baterije od 9 V ili od 6 x 1.5 V - battery holder

- pripadna baterija ili više njih, ovisno o držaču baterije
- dosta žica (jump wires, dupont wires)
- pin headeri - nije nužno, ali pojednostavljuje sastavljanje i urednije izgleda

Od alata potrebna je lemilica jer ću pin headere zalemiti na LCD i RTC radi lakšeg spajanja. Žice se mogu zalemiti i direktno na komponente, ali mi to ne izgleda uredno. Multimetar dobro dođe, ali nije nužan za ovaj projekt. Od softwera potreban je Arduino IDE za izradu sketcha.

Malo detalja o korištenim komponentama

Korišteni LCD i RTC moduli, ali i senzor temperature zaslužuju više pažnje jer o njima dalje ovisi izgled scheme i u konačnici određeni detalji Arduino sketcha.

LCD

U projektu koristim 1602 LCD modul koji je dosta standardan, ima dosta primjera za njega i demo sketcheva, ali njegov je problem što ima puno pinova koji nisu skroz jasni pa bolje pojasniti čemu služe i koje ću od njih koristiti u sketchu. Modul izgleda kao na slici:

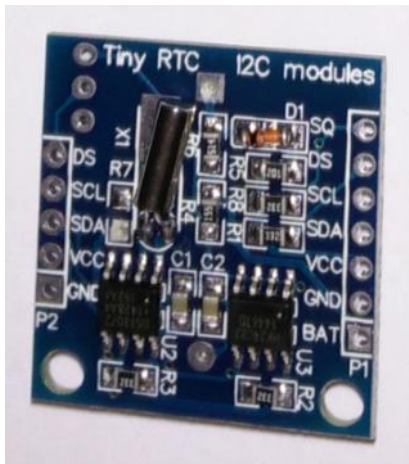


Na modulu su vidljive oznake 1. i 16. pina te oznake svakog pojedinačnog pina. Što pojedini pin predstavlja najbolje je provjeriti u datasheetu modula. Za potrebe projekta koristim datasheet na [linku](#).

Sve sam pinove zalemio na pin headere radi lakšeg daljnjeg sastavljanja. Od svih pinova koristim 1 - 6 i 11 - 16. Pinovi 1 (VSS), 5 (RW) i 16 (K) biti će spojeni na GND dok će pin 2 (VDD) biti spojen na 5 V. Na pin 3 (V0) spojiti ću regulaciju osvjetljenja displaya dok će druga regulacija koja se odnosi na kontrast biti spojena na pin 15 (A). Za upravljanje i podatke dakle preostaju ostali korišteni pinovi: 4 (RS), 6 (E), 11 (D4), 12 (D5), 13 (D6) i 14 (D7). Oni će se direktno spajati na Arduinove pinove zadužene za upravljanje 1602 LCD modulom.

RTC

Korišten RTC (Real Time Clock) modul je dosta zanimljiva komponenta. Koristim ga radi očitavanja točnog vremena, a inicijalni datum i vrijeme modula biti će postavljeni prema datumu i vremenu compilea sketcha koji se odmah uploada na Arduino pločicu. Na netu ima različitih RTC modula, svi su slični, a radi lakšeg pronalaženja na eBayu može se pretraživati po opisu "DS1307 real time clock module Tiny RTC I2C Module". Onaj koji ću koristiti izgleda kao na slici:



Modul sadrži nekoliko pinova, a na svojoj pozadini ima čak i vlastiti držač baterije koji neću koristiti jer će se napajati preko Arduina. Kao što je na slici vidljivo, pinovi se nalaze na dvije suprotne strane, a koristit ću onu s manje njih (na slici lijeva strana s pet pinova). Za korišten Tiny RTC potrebno je pronaći i datasheet, za potrebe projekta koristim datasheet na [linku](#).

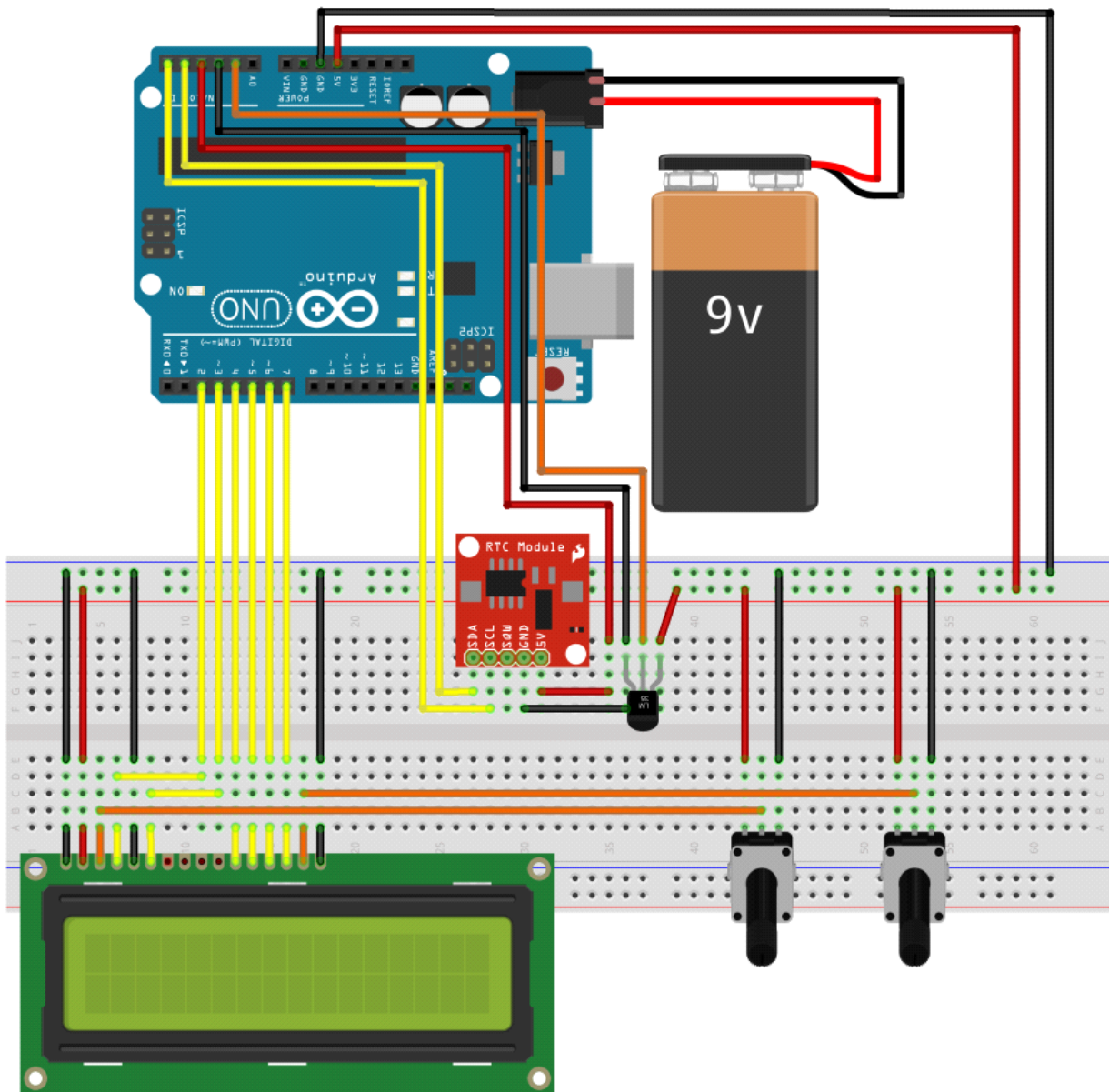
Sve pinove na korištenoj strani sam zalemio na pin headere kako bi se komponenta mogla direktno spojiti na breadboard. Iako su zalemljeni svi pinovi, koristit ću njih četiri: GND, VCC, SDA i SCL. Oni će biti direktno spojeni na Arduino pri čemu SDA mora obavezno biti spojen na Arduino analogni pin A4, a SCL na analogni pin A5.

LM35 senzor temperature

U projektu sam koristio LM35 senzor temperature koji sam dobio u starter kitu. Ima tri pina što je dovoljno, ali je kod spajanja na breadboard uvijek potrebno provjeriti koji što predstavlja. Nikako se ne smiju krivo spajati GND i VCC pinovi na Arduino. Radi provjere pozicija pinova treba provjeriti datasheet komponente, koristio sam ovaj na [linku](#). U nedostatku LM35 senzora temperature, može se koristiti i neki drugi sličan, bitno da je analogan radi jednostavnosti programiranja. U krajnjoj liniji, može poslužiti i digitalni samo bi se za njega morao koristiti pripadni Arduino library.

Skiciranje projekta prije spajanja

Pazeći na raspored pinova gornjih komponenti i količinu žica kojima ćemo ih povezati na Arduino, inicijalna skica projekta u Fritzingu izgleda prema sljedećem:

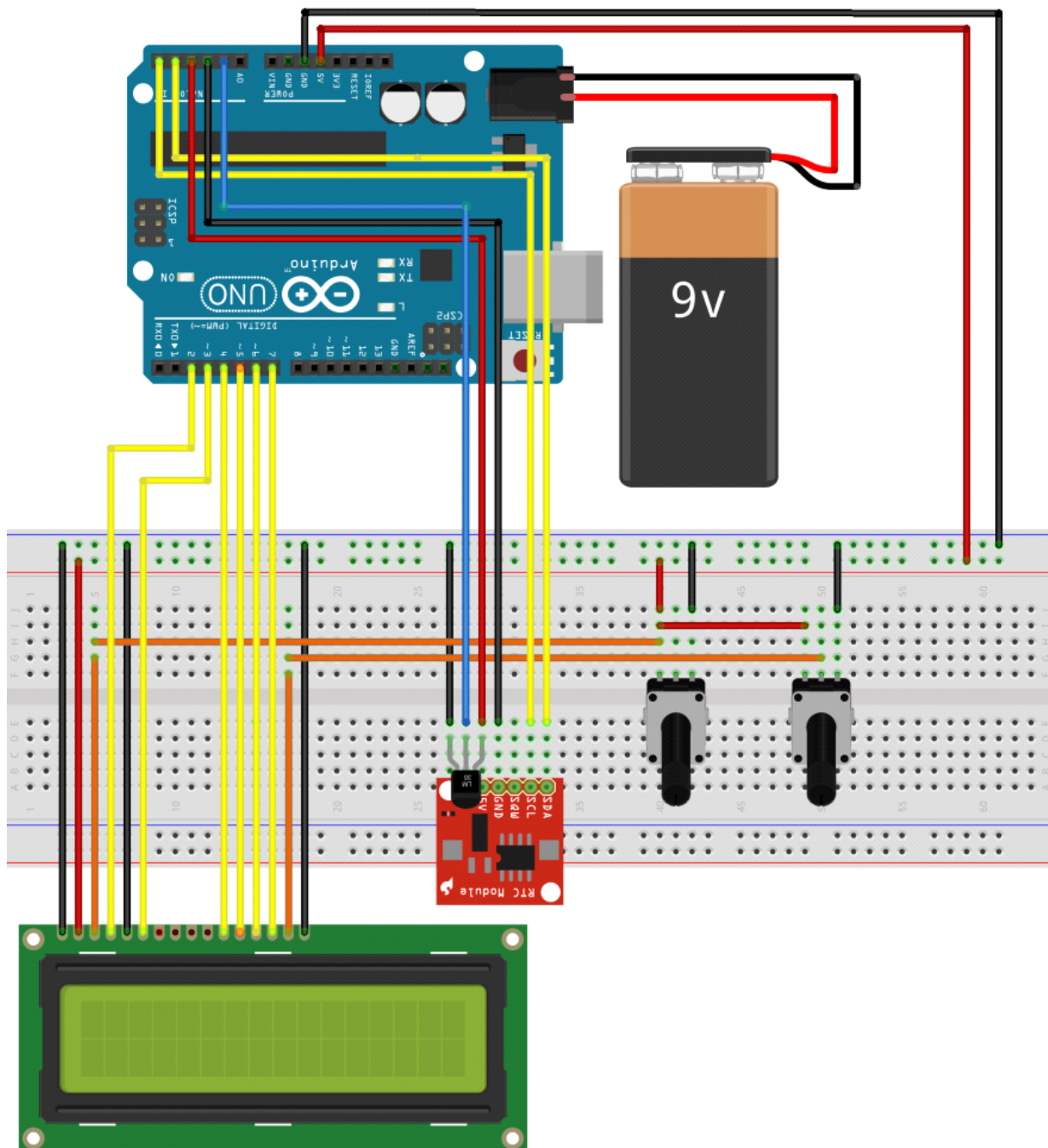


Napomenu bi da simbol RTC modula korištenog u Fritzingu izgleda nešto drugačije od Tiny RTC-a korištenog u stvarnosti, ali to neće utjecati na daljnji tijek projekta. Ovo je početna skica koja bi trebala pregledno predstaviti izgled projekta.

Ako se malo prouči ova schema, vidljivo je da sam VCC i GND pinove RTC modula direktno spojio na analogne pinove Arduina umjesto na (+) i (-) linije pinova na breadboardu. Razlog tome je što sam želio malo optimizirati raspored žica jer ću RTC modul spojiti sa stripeom spojenih žica što mi djeluje vizualno urednije.

Daljnji korak optimizacije bio bi korištenje što manjeg broja drugih žica, pogotovo za spajanje 1602 LCD-a. To međutim za 1602 LCD nije moguće, ali radi boljeg rasporeda 1602 LCD ću spojiti također sa stripeom spojenih žica na Arduino. Zbog toga na Arduino za 1602 LCD koristim digitalne pinove 2 - 7.

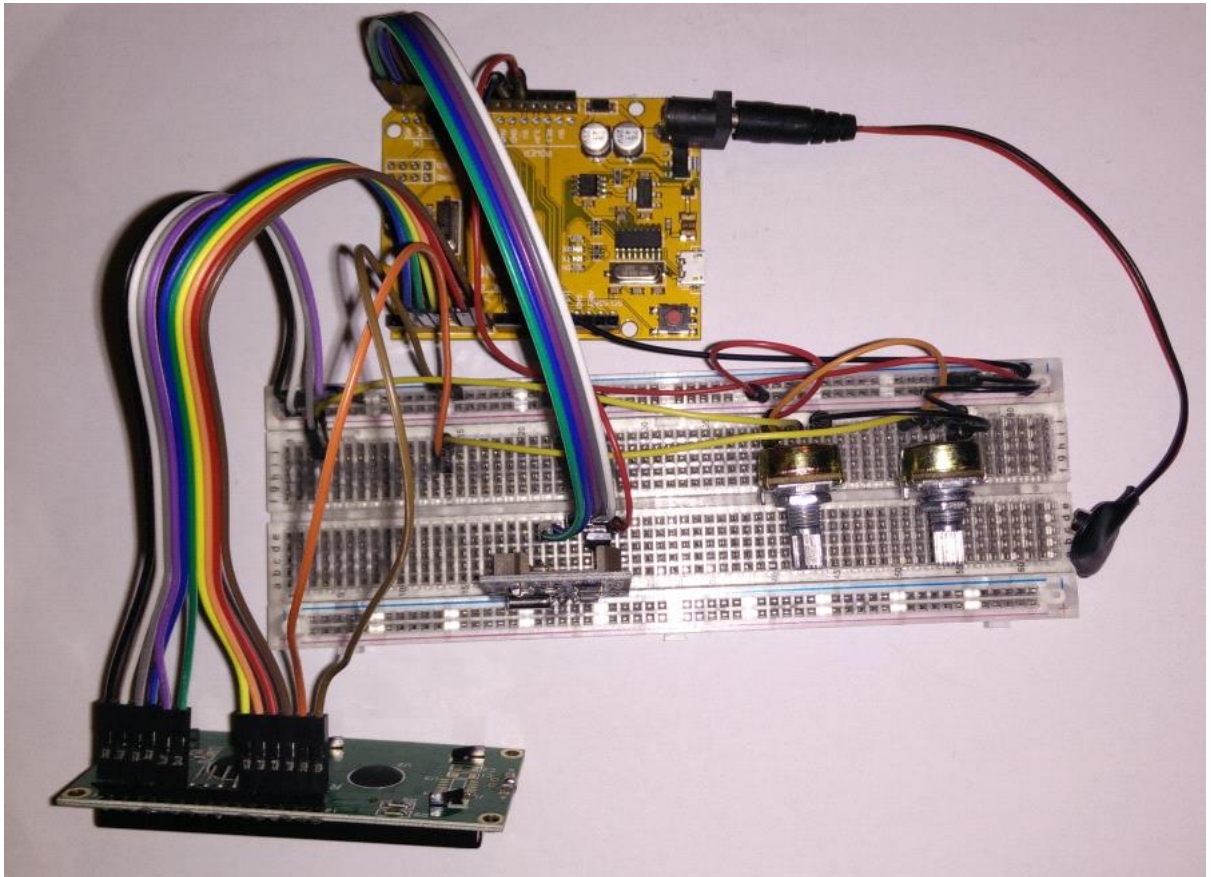
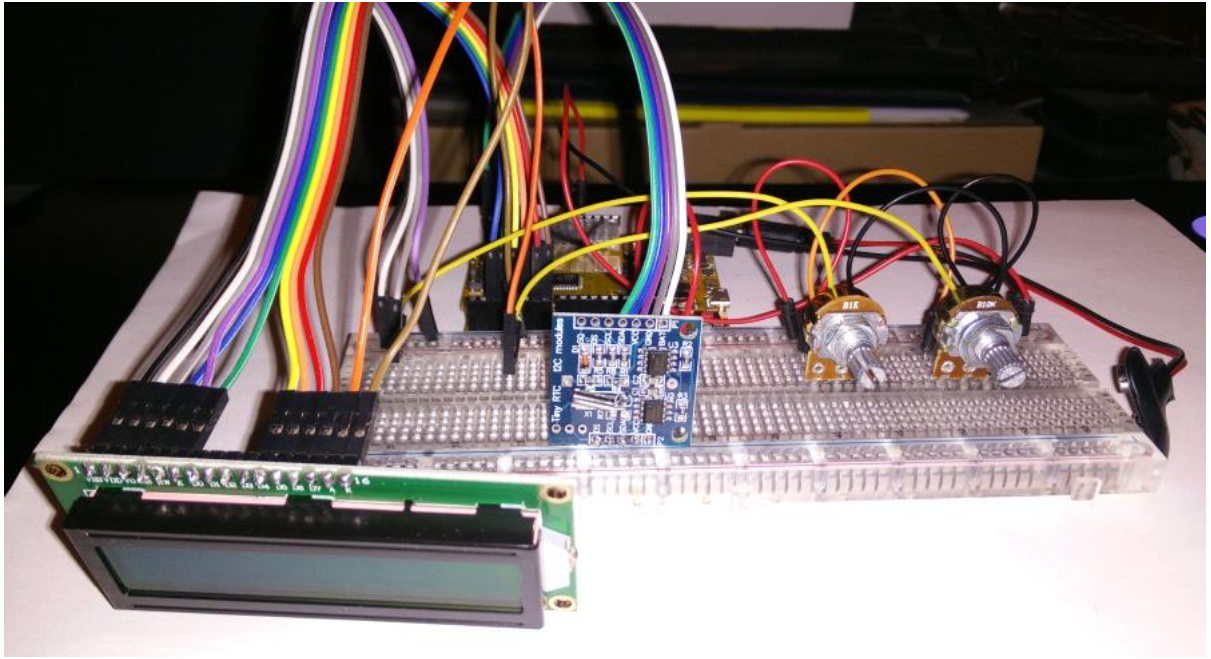
Ova se skica može još malo zakomplicirati, a radi još većeg smanjenja količine žica. Iduća skica prije samog spajanja komponenti je rezultat krajnjeg optimiziranja:

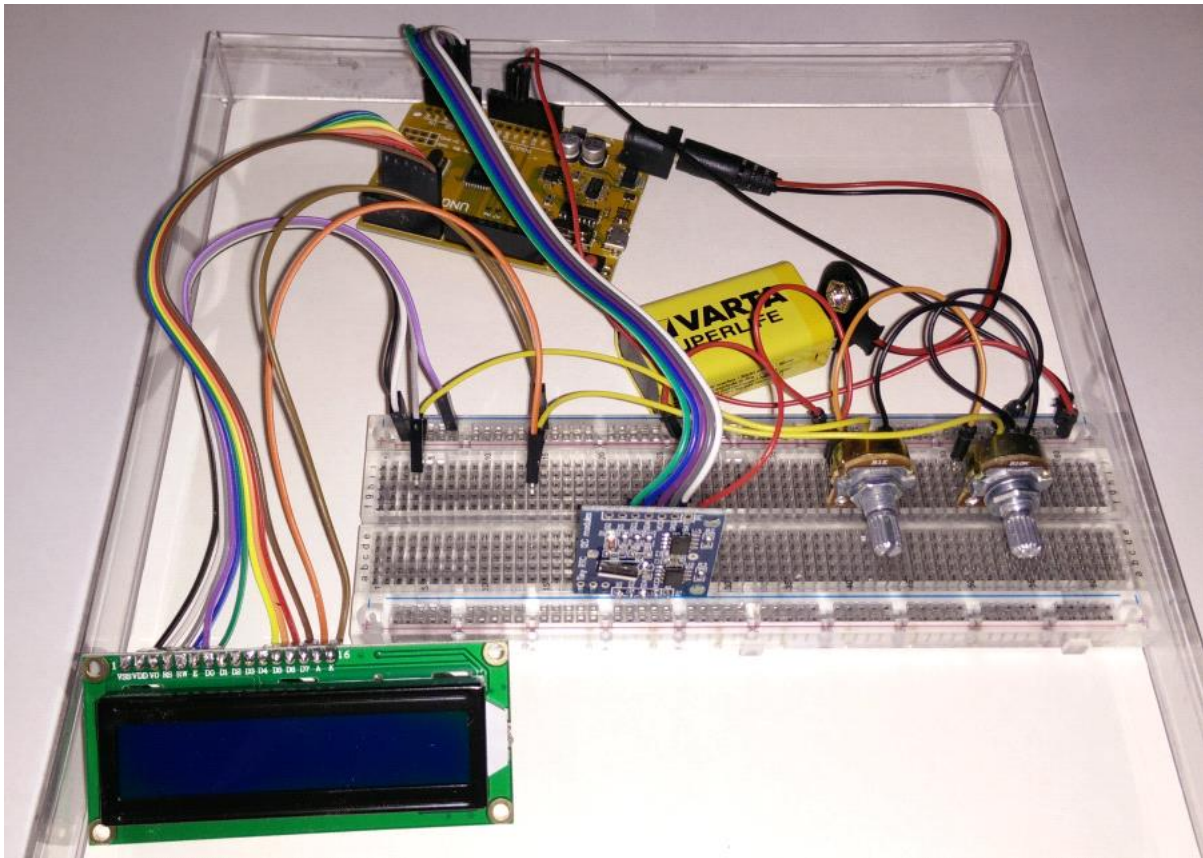
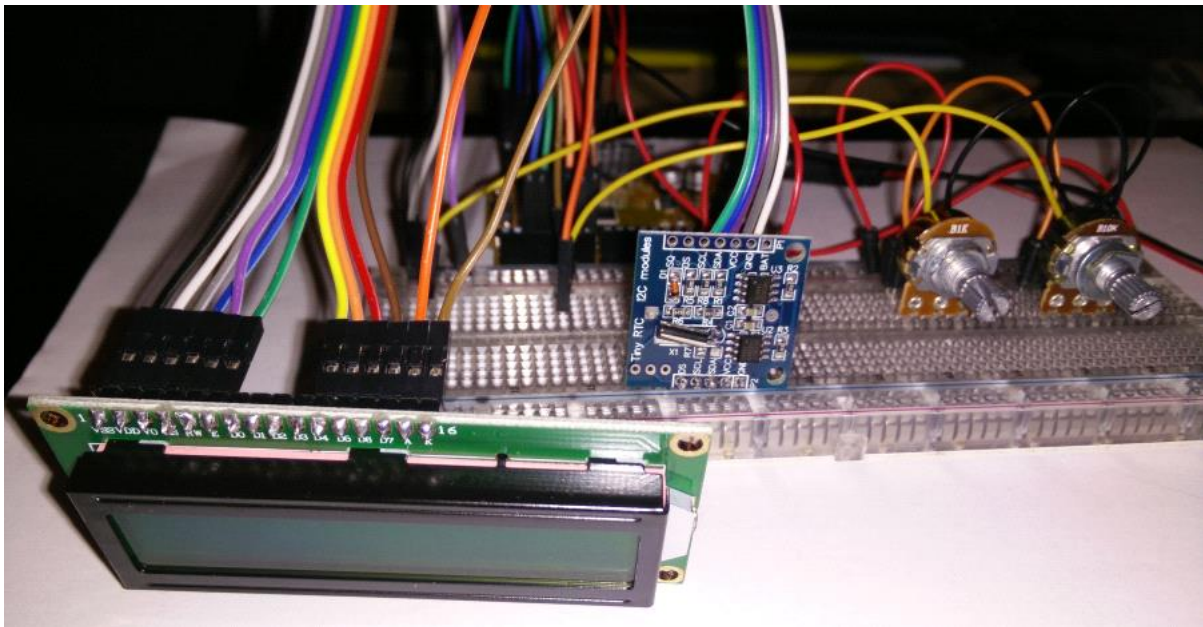


Može se uočiti direktan spoj 1602 LCD-a na Arduino i dijeljenje iste žice VCC-a (5 V) između Tiny RTC-a i LM35 senzora. Na skici se koristi ista žica VCC-a za RTC i LM35 zbog preglednosti i samog rasporeda pinova simbola RTC-a dok ću u praksi, odnosno konkretno u ovom projektu zbog stvarnog rasporeda pinova RTC-a i orijentacije LM35 senzora zapravo dijeliti istu GND žicu. Manje bitno, ali također je vidljiv detalj dijeljenja žice napona između dva potencijometra. Na skici su dva ista potencijometra, ali koristim jedan od 1 k Ω prema pinu 3 LCD-a i jedan od 10 k Ω prema pinu 15 LCD-a. Mogu se koristiti i potencijometri istog otpora te drugih vrijednosti u odnosu na ove koje koristim. Dalje slijedi spajanje svih komponenti u cjelinu.

Skupljanje LCD sata u cjelinu

Prema skici iz prethodnog poglavlja, vrijeme je za sastavljanje projekta. 1602 LCD i Tiny RTC sam zalemio na pin headere radi lakšeg spajanja na breadboard i Arduino. Potencijometar od 1 k Ω spojen je na pin 3 regulacije osvjetljenja 1602 LCD-a dok je potencijometar od 10 k Ω spojen na pin 15 regulacije kontrasta LCD-a. Sastavljen projekt s ponekim detaljem može se vidjeti na priloženim slikama:





Ovako sastavljen projekt još se ne može nazvati Arduino LCD satom, ali ću ga pretvoriti uz nešto programiranja kroz Arduino IDE-u.

Vrijeme je za programiranje

U nastavku ću malo objasniti Arduino sketch koji koristim u ovom projektu LCD sata. Za potrebe sketcha koristim library RTCLib koji ne dolazi s Arduino IDE-om pa ga je potrebno zasebno skinuti i instalirati u folder u kojem se nalaze Arduino libraryji. RTCLib se može preuzeti s [lokacije](#) na kojoj se nalazi [RTCLib-master.zip](#).

Arduino sketch projekta nalazi se u nastavku:

```

/ *****
*
*
*

```

```

*   Arduino LCD sat                                     *
*   *                                                   *
*   Autor: Darko Golner, 2016.                         *
*   *                                                   *
*****/

// library korišten za 1602 LCD modul
#include <LiquidCrystal.h>

// library korišteni za RTC modul
#include <Wire.h>
#include <RTClib.h>

// konstante napona prema 1602 LCD datasheetu
// minimalan napon LCD-a prema datasheetu - Supply Voltage For Logic - Min
#define MIN_VOLTAGE 4.5
// radni napon LCD-a prema datasheetu - Supply Voltage For Logic - Typ
#define MAX_VOLTAGE 5.0

/* korišteni pinovi za interface prema 1602 LCD modulu:
- LCD RS pin to digital pin 2
- LCD Enable pin to digital pin 3
- LCD D4 pin to digital pin 4
- LCD D5 pin to digital pin 5
- LCD D6 pin to digital pin 6
- LCD D7 pin to digital pin 7
- LCD R/W pin to ground
- LCD VSS pin to ground
- LCD VCC pin to 5V
*/
// inicijalizacija libraryja s brojevima pinova za interface LCD-a
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

// RTC modul
RTC_DS1307 RTC;

// pinovi VCC i GND za RTC modul
int RTC_VCC_PIN = A3;
int RTC_GND_PIN = A2;

// pin LM35 senzora temperature
int LM35_OUT_PIN = A1;

// true radi debugiranja kroz Serial Monitor
bool debug = false;

// ostale varijable
int temperature;
int temperatureC;
int counter = 0;
float voltage = 0;
float percent = 0;

// setup programa
void setup() {
  // početak serijske komunikacije
  if (debug) {
    Serial.begin(9600);
  }

  // VCC i GND pinovi za RTC modul
  pinMode(RTC_VCC_PIN, OUTPUT);
  pinMode(RTC_GND_PIN, OUTPUT);
  digitalWrite(RTC_VCC_PIN, HIGH);

```

```

digitalWrite(RTC_GND_PIN, LOW);

// pokretanje RTC modula
Wire.begin();
RTC.begin();
// postavljanje sata na datum i vrijeme compilea programa
RTC.adjust(DateTime(F(__DATE__), F(__TIME__)));
if (debug) {
    if (!RTC.isrunning()) {
        Serial.println("RTC modul ne radi!");
    } else {
        Serial.println("RTC radi!");
    }
}

// LM35 senzor temperature ima referentan napon od 1.1 V
analogReference(INTERNAL);
pinMode(LM35_OUT_PIN, INPUT);

// 1602 LCD modul
lcd.begin(16, 2);
lcd.clear();

delay(1000);
}

// glavna procedura programa
void loop() {
    static char str[16];

    // očitavanje vremena iz RTC modula
    DateTime now = RTC.now();
    if (debug) {
        Serial.print("Datum i vrijeme: ");
        Serial.print(now.day(), DEC);
        Serial.print('.');
        Serial.print(now.month(), DEC);
        Serial.print('.');
        Serial.print(now.year(), DEC);
        Serial.print("  ");
        Serial.print(now.hour(), DEC);
        Serial.print(':');
        Serial.print(now.minute(), DEC);
        Serial.print(':');
        Serial.print(now.second(), DEC);
        Serial.println();
    }

    // LM35 senzor temperature
    // očitavanje temperature na početku i svakih 10 koraka programa
    if (counter == 0 || counter == 9) {
        temperature = analogRead(LM35_OUT_PIN); // očitavanje s pina
        /* analogReference je postavljen na INTERNAL, maksimalan napon je 1.1 V
           senzor mjeri vrijednosti 0 - 1023, a svaki °C odgovara 10 mV
           1 V = 1000 mV koji se dijele s 10 mV radi °C
           u izračunu je 110 zbog 1.1 * 100 da bi dobili V
        */
        temperatureC = round(110 * temperature / 1023);
        if (debug) {
            Serial.println("Vrijednost senzora temperature: " + String(temperature));
            Serial.println("Temperatura u C: " + String(temperatureC));
        }
        counter = 0;
    }
}

```

```

counter++;

// interni pin 0 služi za očitavanje napona
// za izračun se koriste referentni naponi LCD modula
voltage = analogRead(0) * MAX_VOLTAGE / 1023;
// postotak izmjerenog napona u odnosu na maksimalni
percent = round(((voltage - MIN_VOLTAGE) * 100) /
                (MAX_VOLTAGE - MIN_VOLTAGE));
if (percent < 0) {
    percent = 0;
}
if (percent > 100) {
    percent = 100;
}
if (debug) {
    Serial.println("Napon: " + String((int)voltage));
    Serial.println("Baterija: " + String((int)percent) + "%");
}

// ispis vremena i datuma u prvom redu LCD-u
lcd.setCursor(0, 0);
sprintf(str, "%02d:%02d:%02d  %02d.%02d.",
        now.hour(), now.minute(), now.second(),
        now.day(), now.month());
lcd.print(str);
// ispis temperature i postotka baterije u drugom redu LCD-a
lcd.setCursor(0, 1);
lcd.print("Temp:");
sprintf(str, "%02d", temperatureC);
lcd.print(str);
lcd.print((char)223); // ASCII znak za stupanj
lcd.print('C');
if (percent > 0) {
    sprintf(str, " B:%3d%", round(percent));
    lcd.print(str);
}

delay(1000);
}

```

Unutar sketcha nalazi se nekoliko bitnih detalja. Jedan od njih odnosi se na konstante minimalnog i tipičnog napona 1602 LCD-a prema datasheetu komponente (ako vaša komponenta ima drugačije vrijednosti, potrebno ih je izmijeniti u sketchu):

```

// minimalan napon LCD-a prema datasheetu - Supply Voltage For Logic - Min
#define MIN_VOLTAGE 4.5
// radni napon LCD-a prema datasheetu - Supply Voltage For Logic - Typ
#define MAX_VOLTAGE 5.0

```

Ove vrijednosti koristim radi izračuna i prikazivanja preostalog postotka kapaciteta korištene baterije. Za pitanje kakva je veza između 1602 LCD-a i kapaciteta baterije, postoji objašnjenje. Da bi znao koliko je baterije preostalo, koristit ću Arduinov interni voltmetar, skriven feature Arduina koji mogu dobiti analognim očitanjem vrijednosti napona na pinu 0. Maksimalna vrijednost očitavanja na analognom pinu je 1023 i ona predstavlja 5 V što bi predstavljalo da je baterija na kapacitetu od 100 %. Kako LCD prema datasheetu radi na minimalno 4.5 V, taj će mi napon predstavljati 0 % kapaciteta baterije jer u tom slučaju čim napon padne ispod te vrijednosti LCD ne radi. Dalje samo preostaje izračun postotka prema izmjerenom naponu u odnosu na raspon napona pod kojim radi LCD. Treba naglasiti da ovo ne mora biti nužno pouzdana metoda provjere preostalog kapaciteta baterije.

Drugi detalj odnosi se na definiciju koji se pinovi koriste za interface prema 1602 LCD-u. Upravljački pinovi LCD-a i pripadni pinova Arduina u inicijalizaciji moraju odgovarati onako kako su fizički spojeni, zbog toga je dodan komentar prije inicijalizacije:

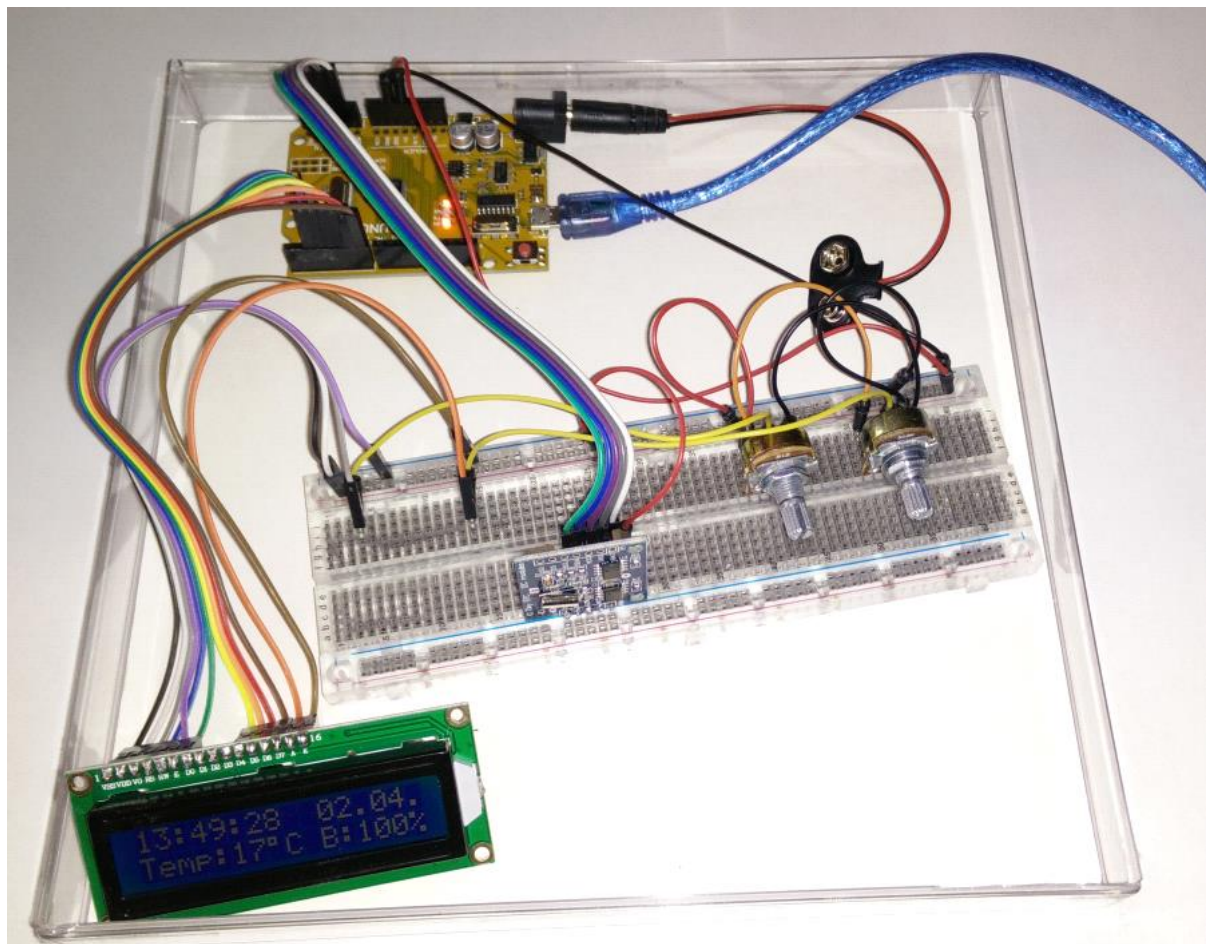
```

/* korišteni pinovi za interface prema 1602 LCD modulu:
- LCD RS pin to digital pin 2
- LCD Enable pin to digital pin 3
- LCD D4 pin to digital pin 4
- LCD D5 pin to digital pin 5
- LCD D6 pin to digital pin 6
- LCD D7 pin to digital pin 7
- LCD R/W pin to ground
- LCD VSS pin to ground
- LCD VCC pin to 5V
*/
// inicijalizacija libraryja s brojevima pinova za interface LCD-a
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

```

Treći detalj vezan je za definiranje VCC i GND pinova Tiny RTC-a. Zbog načina spajanja žica definirao sam da će mi analogni pin A3 biti VCC, a analogni pin A2 bit će GND. Ostale pinove RTC-a sam ranije opisao, SDA mora obavezno biti spojen na analogni pin A4, a SCL na analogni pin A5.

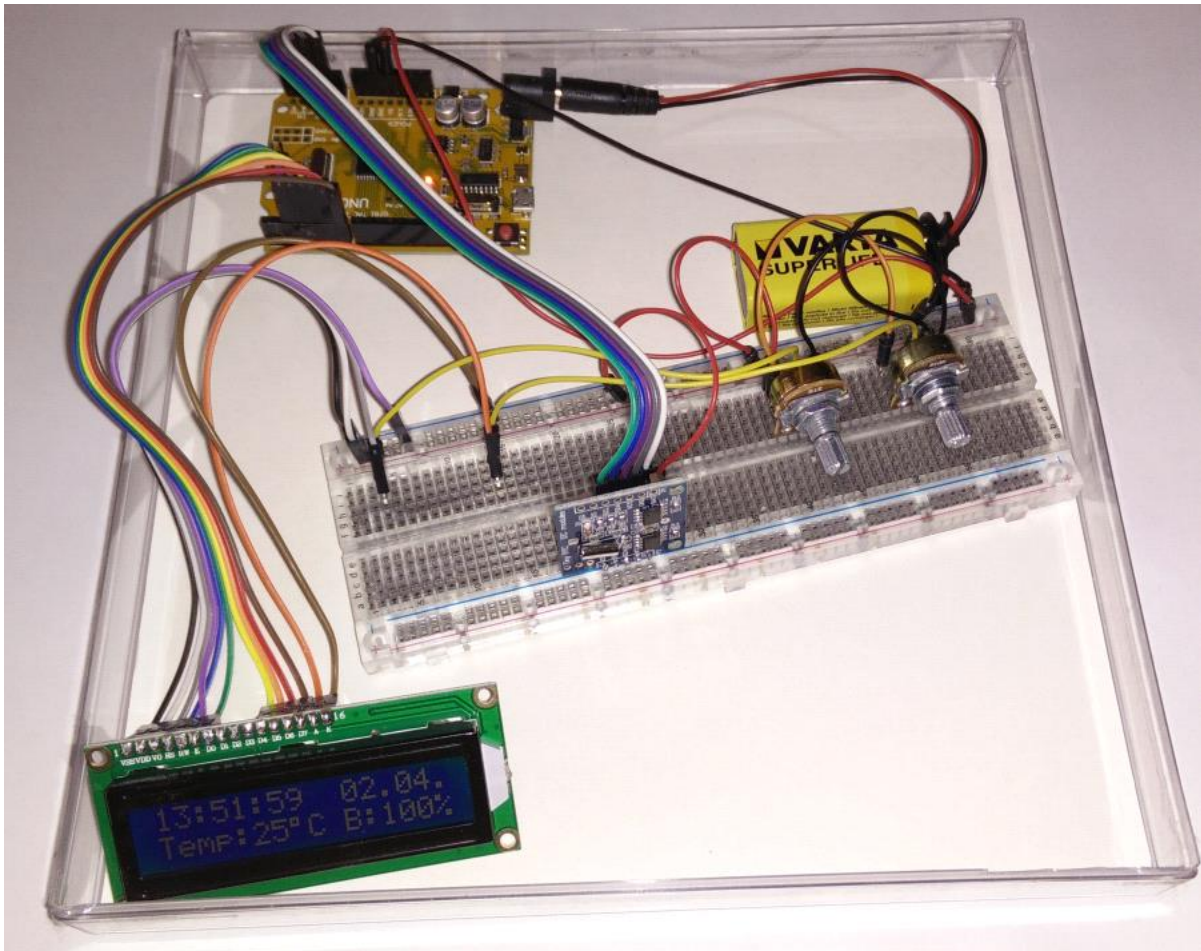
Nadalje se sketch može compilirati i potom odmah uploadati na Arduino pločicu. Ukoliko je sve ispravno, sat bi trebao raditi, odnosno trebalo bi se vidjeti točno vrijeme, datum, temperatura u °C i preostali kapacitet baterije (kad je na USB-u, prikazuje se 100 %). Za početak najbolje odmah provjeriti da li radi dok je još uvijek na USB kabelu:



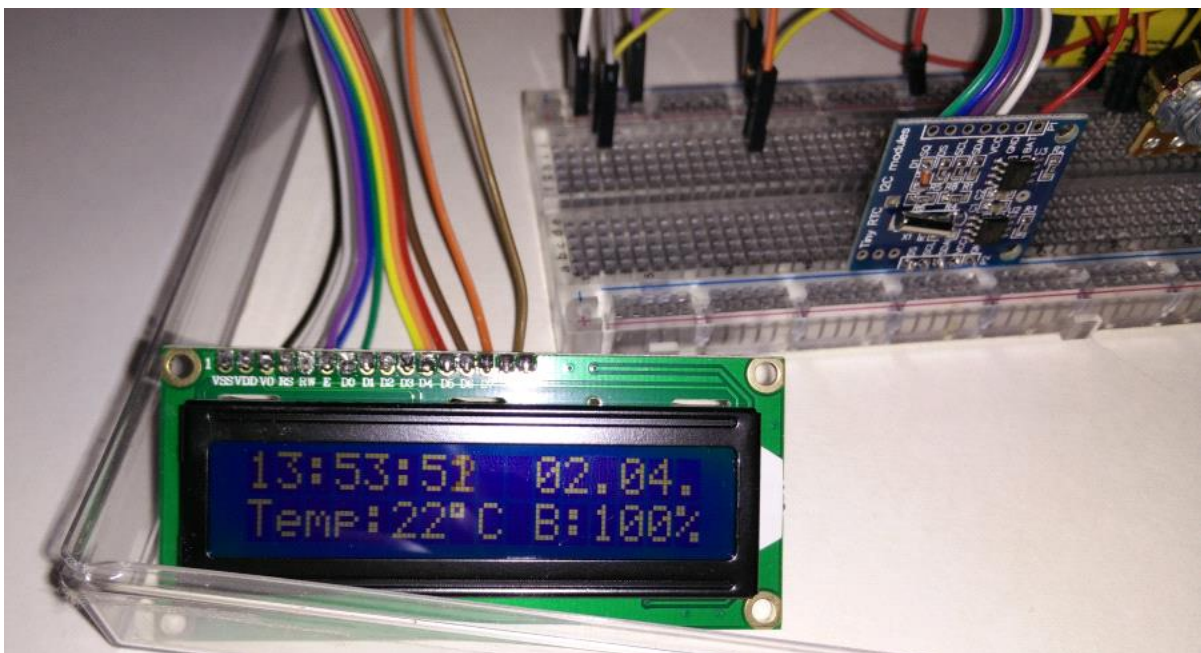
U slučaju da ništa nije vidljivo, okretanjem potenciometra može se provjeriti da li se radi o pre niskom intenzitetu osvjetljenja ili kontrasta zbog kojeg se ništa ne vidi. Za druge slučajeve kada nešto ne radi, najbolje je u sketchu postaviti vrijednost varijable za debug u true i pratiti što je sve vidljivo u Serial Monitoru.

Puštanje u živi rad

Ukoliko je sve prošlo ispravno, može se spojiti baterija i odspojiti USB kabel:



Dalje samo treba pratiti da li sve funkcionira ispravno:



Dodatno, kako pobliže izgleda projekt Arduino LCD sata s namještanjem intenziteta osvjetljenja i kontrasta može se vidjeti u videu:

<https://youtu.be/dqx3qGnUdXk>

Download materijali

[PDF tutorial](#)

[lcd_sat.ino](#)

Dodatni linkovi

Tiny RTC

http://www.electrow.com/wiki/index.php?title=Tiny_RTC

Arduino analogReference

<https://www.arduino.cc/en/Reference/AnalogReference>

Arduino interni voltmetar

<https://www.arduino.cc/en/Tutorial/ReadAnalogVoltage>

Objavljeno 2016. g.