

## Arduino i nadzor parametara prostorije



Seriju članaka o malenim, ali moćnim platformama, zaokružiti ćemo Arduinoom pomoću kojeg ćemo mjeriti temperaturu i vlažnost prostorije, te količinu svjetla.

Arduino je razvijen u Italiji za poučavanje programiranja u C/C++ programskim jezicima. Napravljen je od *open source* hardverskih komponenti koje su spojene na 8-bitni Atmel AVR mikrokontroler ili 32-bitni ARM procesor. Trenutno postoji 16 različitih modela, a svi se mogu proširivati sa tzv. *shield*-ovima, gotovim PCB-ovima koji imaju različite funkcije. Arduino dolazi u različitim veličinama, na slici 1 prikazani su Arduino Uno i Arduino (Pro) Mini zajedno s jednom AA baterijom, čisto za usporedbu veličine.



Slika 1.

U našem primjeru koristit ćemo model Uno čiji su dijelovi opisani na slici 2. Što se tiče cijene, kineske kopije mogu se pronaći za 9,99 USD, a nude istu funkcionalnost kao i talijanski originali uz nešto manju kvalitetu izrade.



[1]

Slika 2. (klik za uvećanje)

Arduino se programira putem Arduino Software-a, besplatnim *IDE*-om u kojem se piše C/C++ programski kod i prebacuje na Arduino. On radi na Windows platformi, MAC OS X-u i Linuxu, a može se doći i do izvornog koda. S *IDE*-om dolazi velik broj primjera koji mogu dobro poslužiti za upoznavanje s programiranjem.



Slika 3.

Klikom na prvu ikonu gore lijevo (znak "viđeno") obavlja se kompajliranje, a sljedeće dugme (sa strelicom udesno) prebacuje program na Arduino koji je na računalo spojeno USB kabelom. Koliko je programiranje jednostavno možete zaključiti iz priloženog programskog koda koji naizmjenično pali i gasi jednu ledicu spojenu na Arduino. Ovo je jedan od primjera iz *IDE*-a:

```
int led = 5;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

Petlja unutar programa na digitalni izlaz (pin 5) svake sekunde šalje naizmjenično signale HIGH i LOW, što u prijevodu znači da se ledica pali i gasi.

Program se pokreće odmah po prebacivanju na Arduino. Ovisno o tome što radi, rezultati su vidljivi (primjerice, na nekoj vizualnoj komponenti poput ekrana) ili je potrebno uključiti *Serial Print* (pod opcijom *Tools*) kojim može uživo pratiti rezultate programa koji se negdje zapisuju. Arduino se uključuje odmah po detekciji napona i program se, ako postoji, počinje automatski izvršavati, što je zgodno pri nestanku struje. Također, ne postoji tipka za gašenje, dovoljno je isključiti napajanje.

Spomenuli smo *shield*-ove, kojima se svaki Arduino može proširiti. Trenutno postoji više od 300

*shield*-ova od 125 različitih proizvođača, a osim gotovih, moguće je napraviti i svoj. Brojke pokazuju da je Arduino vrlo popularan i da ne postoji njegova primjena koja nije riješena na neki način. *Shield*-ovi se mogu slagati jedan na drugog (eng. *stacking*). Na slici 4 su prikazani neki od *shield*-ova, a od zanimljivijih spomenimo servo motore.



[2]

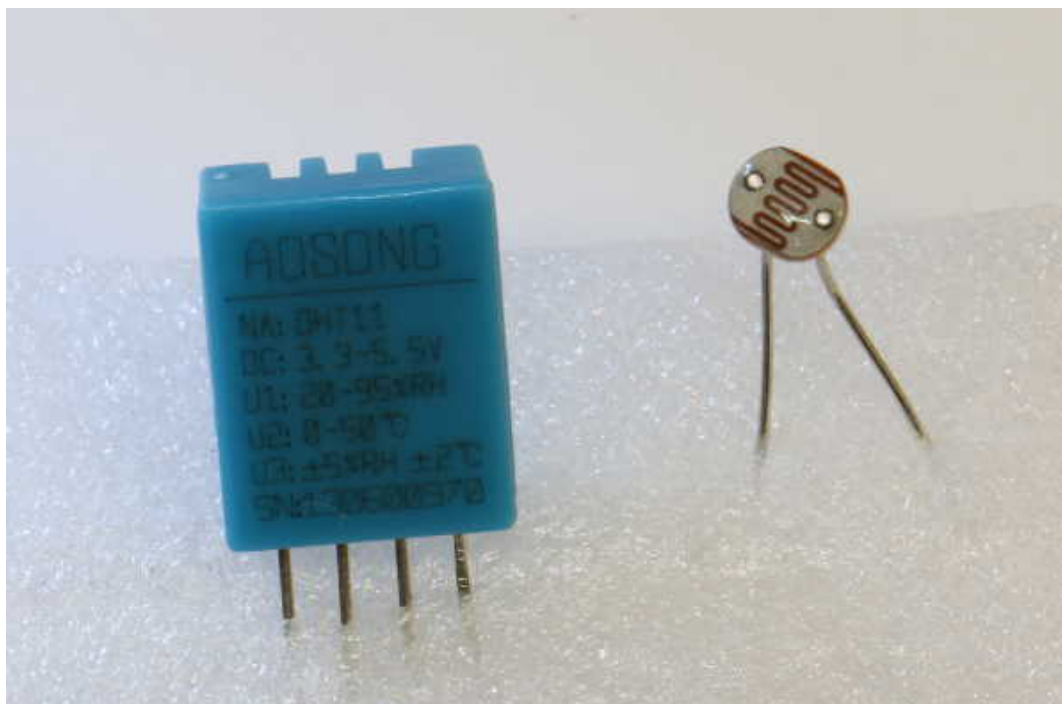
Slika 4. (klik za uvećanje)

Na slici 4 prikazani su *shield*ovi redom: 1 – ekran Nokije 3310 (nije baš originalni shield), 2 – 10BASE-T ethernet shield, 3 – RF primopredajnik, 4 – senzor za razinu vode, 5 – LCD ekran 16x2, 6 – konverter USB na TTL Serial, 7 – ultrazvučni senzor za udaljenost (do 2m), 8 – akcelerometar, 9 – PIR senzor, 10 – IR odašiljač, 11 – žiro senzor, 12 – RF primopredajnik, 13 – shield za SD karticu, 14 – IR prijemnik, 15 – RF prijemnik (433MHz), 16 – breadboard (400 pinova), služi za priključivanje elemenata i shield-ova na Arduino, 17 – RFID čitač/pisač, 18 – RF prijemnik, 19 – numerička tipkovnica

Vratimo se na naš primjer mjerenja temperature i vlažnosti zraka, te količine svjetla. Za to će nam trebati dva senzora, jedan za mjerenje temperature i vlažnosti, a drugi za mjerenje količine svjetla. Osim njih spojiti ćemo i ethernet *shield*. Time smo omogućili spremanje podataka na udaljeno računalo ili nadzor parametara u realnom vremenu. Za spajanje *shield*-ova koristit ćemo *breadboard*, ali sve komponente je moguće staviti tj. zalemiti na *do-it-yourself* pločicu i to sve spakirati kao gotov proizvod. *Breadboard* se uglavnom koristi za testiranje i nije baš preporučljivo postavljati ga trajno u radno okruženje.

Senzor za temperaturu i vlažnost ima oznaku DHT11, a mjeri temperaturu u opsegu od 0 do 50 stupnjeva Celzijusa s preciznošću od  $\pm 2$  stupnja, te vlažnost u opsegu od 20 do 80% s greškom 5%. Iz priloženog je vidljivo da mu preciznost nije jača strana, ali cijena jest, može se naći za nešto više od 1 USD. Senzor radi na 5V, a izlaz koji daje je digitalni. Ako vam je preciznost bitna, možete nabaviti DHT22 koji mjeri temperaturu u opsegu od -40 do +125 stupnjeva Celzijusa uz preciznost 0.5 stupnjeva. On je nešto skuplji, cca 4 USD.

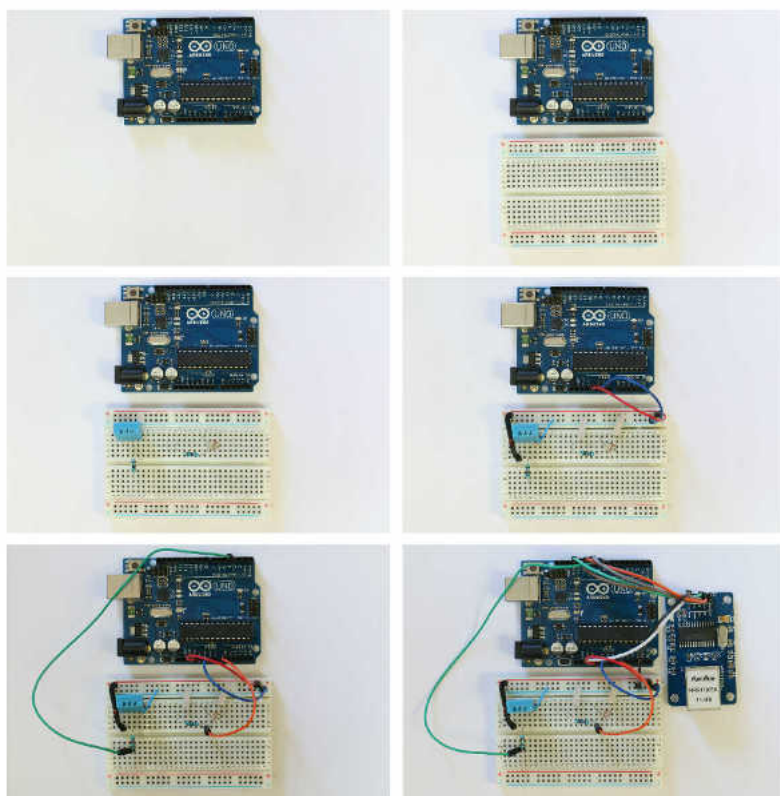
Senzor količine svjetla je, ustvari, fotootpornik (LDR) vrlo malih dimenzija, u promjeru ima samo 5 mm. Raspon otpora je od 200 KOhma (mrak) do 10 KOhma (svjetlost jačine 10 luxa), a osjetljiv je na svjetlo valnih duljina od 400 nm (ljubičasto) do 600 nm (narančasto). On također radi na 5V, a izlaz je analogna vrijednost koja ima raspon vrijednosti od 0 (mrak) do 1023 (jako svjetlo). Dovoljno je osjetljiv da se može koristiti kao alarm za čuvanu sobu u slučaju da "provalnik" ima samo malu ručnu lampu. Cijena mu je otprilike 0,11 USD. Priznat ćete, iznos zvuči smiješno. Na slici 5 su uvećana oba senzora.



Slika 5.

Zadnji shield koji ćemo koristiti je 10BASE-T ethernet *shield* koji radi na 3.3V. Opet, brzina mu nije najjača strana, ali je i više nego dovoljna za našu primjenu. Cijena mu je oko 3,5 USD.

Spajanje komponenti je prikazano u koracima na slici 6. Na početku su prikazani Arduino Uno i *breadboard* bez ikakvih drugih elemenata. Kasnije smo dodavali senzore, otpornike i spajali kabele na odgovarajuće pinove, a na zadnjoj slici je prikazan kompletan sklop zajedno s ethernet *shield*-om (krajnje desno).



[3]

Slika 6. (klik za uvećanje) [3]

Slijedi pisanje programa. Ideja je da senzori svaku sekundu očitaju vrijednosti i prikažu ih na web stranici koju ćemo staviti na Arduino. Time ćemo moći u svakom trenutku pristupiti "stroju" preko mreže i očitati vrijednosti. Naravno, umjesto web stranice podaci se mogu zapisivati na memorijsku karticu (potreban je novi *shield* ili drugi model Arduina) ili na neki drugi poslužitelj.

```
#include <EtherCard.h>
#include <dht.h>
#define dht_dpin A0

dht DHT;
int LDRPin = 1;
int LDR;

static byte mymac[] = { 0x74,0x69,0x69,0x2D,0x30,0x31 };
static byte myip[] = { 192,168,0,5 };
byte Ethernet::buffer[1000];
BufferFiller bfill;

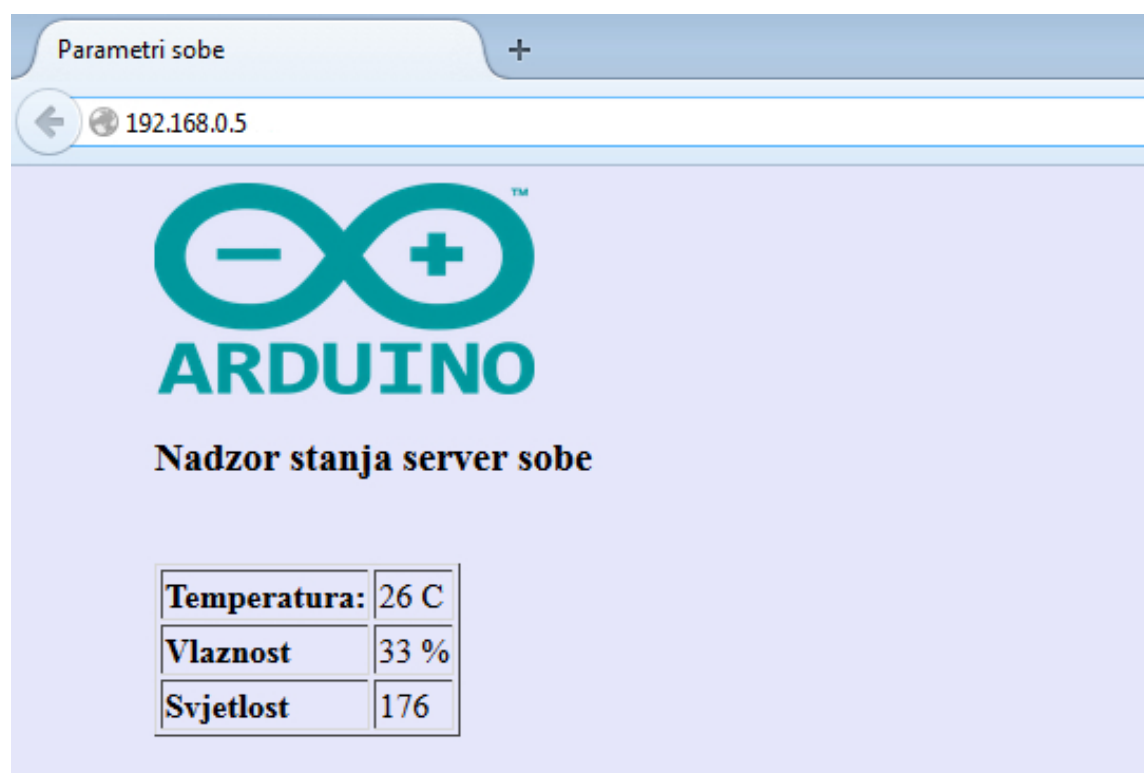
void setup () {
  if (ether.begin(sizeof Ethernet::buffer, mymac) == 0)
    Serial.println( "Failed to access Ethernet controller");
  ether.staticSetup(myip);
}

static word homePage() {
  DHT.read11(dht_dpin);
  LDR = analogRead(LDRPin);
  bfill = ether.tcpOffset();
  bfill.emit_p(PSTR(
    "HTTP/1.0 200 OK\r\n"
    "Content-Type: text/html\r\n"
    "Pragma: no-cache\r\n"
    "\r\n"
    "<meta http-equiv='refresh' content='1' />"
    "<body bgcolor='#E6E6FA'>"
    "<div style='display: block;width:300px;height:100px;top:15%;margin-
left:auto;margin-right:auto>"
    "<img src='http://upload.wikimedia.org/wikipedia/commons/thumb/8/87/Arduino_Logo.s
vg/720px-
Arduino_Logo.svg.png' alt='arduino' width='180' height='100' align='center'>"
    "</div>"
    "<title>Parametri sobe</title>"
    "<h3>Nadzor stanja server sobe</h3>"
    "&nbsp;"
    "<table border='1' color='blue'><tr><td><b>Temperatura:</b></td><td>$D C</td></tr>
<tr><td><b>Vlaznost</b></td><td>$D %</td></tr><tr><td><b>Svjetlost</b></td><td>$D</td>
</tr></table>"
    "&nbsp;"
  ),

  byte(DHT.temperature),
  byte(DHT.humidity),
  byte(LDR));
  return bfill.position();
}
```

```
void loop () {  
  word len = ether.packetReceive();  
  word pos = ether.packetLoop(len);  
  
  if (pos)  
    ether.httpServerReply(homePage());  
}
```

Kako programski jezici C/C++ nemaju naredbe za HTML, potrebno je učitati biblioteku koja to omogućava. Međutim, ne radi se o potpunom HTML-u, a i samo pisanje tagova nije zgodno kao u "pravom" HTML-u, što je vidljivo iz primjera. Na početku smo uključili potrebne biblioteke. Ethernet *shield*, za razliku od klasičnih mrežnih kartica, nema tvornički upisanu MAC adresu pa je nju potrebno ručno postaviti, kao i IP adresu. Nakon provjere da li je ethernet *shield* uopće priključen, kreiramo HTML tablicu i u njoj ispisujemo parametre koje dobivamo od senzora. Glavni dio programa u beskonačnoj petlji poziva prikaz stranice ukoliko postoje podaci koji dolaze sa ethernet priključka, a stranica se osvježava svake sekunde. Prilično jednostavno, zar ne? Rezultat programa je web stranica prikazana na slici 7.



Slika 7.

Nismo naveli najzgodniju stvar, a to je izbacivanje samog Arduina iz igre. Naime, program je moguće "spržiti" na mikrokontroler (primjerice, ATMEGA 328P-PU) kojeg onda smjestimo na pločicu zajedno s kristalom koji daje takt (16 MHz) i dva keramička kondenzatora (0,1 pF). Na istu pločicu spojimo *shield*-ove i ostale elemente, dovedemo napajanje i tako dobijemo neovisni sustav, a Arduino nam ostaje slobodan za novu igru.

Načine primjene Arduina ograničava samo vaša mašta. Primjerice, korisnim se može pokazati IR odašiljač kojim je moguće upaliti klima uređaj kad struja nestane i vrati se. Ili s IR prijemnikom napraviti daljinski upravljač za otvaranje garaže. Još jedna "luda" ideja je samostalni sustav za zalijevanje cvijeća. A naš sustav iz primjera se lako može nadograditi da glumi pravu meteorološku stanicu. Kad krenu, nove ideje same naviru, a sve za iznos koji se lako potroši u jednom večernjem izlasku. Sustav koji smo naveli u primjeru ukupno košta 100-njak kuna, a zabava uz učenje je neprocjenjiva.

Vezani članci:

[Raspberry Pi i Arduino - 1. dio](#) [4]

[Raspberry Pi i video nadzor](#) [5]

pet, 2014-07-04 12:57 - Mirko Lovričević **Kategorije:** [Hardware](#) [6]

[Arduino](#) [7]

**Vote:** 5

Vaša ocjena: Nema Average: 5 (1 vote)

**Source URL:** <https://sysportal.carnet.hr/node/1418>

#### Links

[1] <https://sysportal.carnet.hr/system/files/slika2.JPG>

[2] <https://sysportal.carnet.hr/system/files/slika4.JPG>

[3] [https://sysportal.carnet.hr/system/files/slika6\\_0.jpg](https://sysportal.carnet.hr/system/files/slika6_0.jpg)

[4] <https://sysportal.carnet.hr/node/1381>

[5] <https://sysportal.carnet.hr/node/1392>

[6] <https://sysportal.carnet.hr/taxonomy/term/24>

[7] <https://sysportal.carnet.hr/taxonomy/term/65>