

OSTVARIVANJE SUČELJA IZMEĐU MIKROUPRAVLJAČA I MATLAB-a

IMPLEMENTING INTERFACE BETWEEN MICROCONTROLLER AND MATLAB

Bruno Palašek, Petra Mesarić, Mihael Kukec

Stručni članak

Sažetak: Temperatura i vlažnost zraka dvije su najčešće mjerene atmosferske veličine, što se može i očekivati s obzirom da je većina fizičkih, elektronskih, kemijskih, mehaničkih i bioloških sustava pod utjecajem temperature i vlažnosti. U radu je prikazan primjer mjerenja temperature, vlažnosti i intenziteta svjetlosti uz pomoć mikroupravljača. Izmjerene vrijednosti u realnom vremenu, šalju se na računalo serijskom komunikacijskom vezom te analiziraju i prikazuju uz pomoć Matlab-a. Prikazano je i razvojno okruženje Arduino koje se koristi za mjerenje fizikalnih veličina, a jedno je od najpopularnijih razvojnih platformi u današnjem akademskom i inženjerskom svijetu.

Ključne riječi: analiza podataka, Matlab, Arduino, mikroupravljač, senzori

Professional paper

Abstract: Temperature and humidity are two most commonly measured atmospheric values as can be expected, since most of physical, electronic, chemical, medical and biological systems are influenced by them. This paper presents an example of measuring temperature, humidity and light intensity realized by microcontroller. Real time measured values are sent to the computer by serial communication link then analyzed and displayed in MATLAB. The paper also presents Arduino development environment, used for measuring physical variables. Thus it is represented as one of the most popular development platforms in today's academic and hobby engineering world.

Key words: data analysis, Matlab, Arduino, microcontroller, sensors

1. UVOD

Praćenje rada sustava na temelju prikupljenih podataka može se postignuti dodatnim integriranjem senzora. Takav podsustav za prikupljanje podataka u realnom vremenu uključuje: mjerenje signala realnog fizičkog sustava uz pomoć različitih senzora, digitalizaciju signala za pohranu, analizu te prezentaciju na računalo. Koristeći različite senzore moguće je mjeriti fizičke fenomene kao što su temperatura, intenzitet svjetla, pritisak, vlažnost, sila ili neka druga veličina.

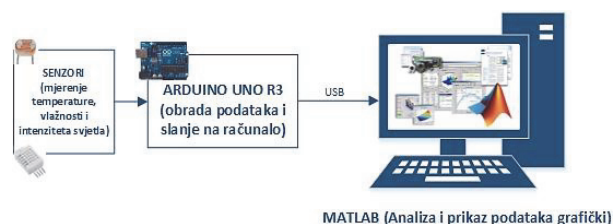
Kada neki sustav ne radi kako bih trebao, inženjerima nedostatak potrebnih podataka predstavlja zahtjevnije rješavanje problema. Kroz sustav koji se temelji na sensorima i prikupljanju podataka u realnom vremenu problem može biti riješen u kratkom vremenu. Prikupljanje podataka postaje važno, stoga je u ovom radu dan primjer temeljen na mikroupravljaču koji mjeri fizikalne veličine u realnom vremenu i prikazuje mjerene veličine uz pomoć Matlab-a (engl. *Matrix Laboratory*). Programski jezik Matlab namijenjen je za tehničke proračune te tako objedinjava računanje, vizualizaciju i programiranje u lako uporabljivoj okolini, u kojoj su problem i rješenje definirani poznatom matematičkom notacijom [1].

Mikroupravljač je u svojoj osnovnoj izvedbi cijeli računalni sustav sadržan u jednom integriranom krugu. Jedan od osnovnih izazova u razvoju nekog sustava temeljenog na mikroupravljaču jest odabir najboljeg

mikroupravljača za određeni dizajn. Cilj je odabrati optimalni mikroupravljač koji ima potrebne parametre i mogućnosti [2]. U ovom radu za mjerenje temperature, vlažnosti i intenziteta svjetlosti koristi se mikroupravljač Atmel ATmega328P koji je integriran u sklopu Arduino razvojnog okruženja.

2. SOFTVER SUSTAVA ZA MJERENJE FIZIKALNIH VELIČINA

Prikupljanje podataka sa senzora uz pomoć Matlab-a izvedeno je pomoću osnovnih alata unutar samog program. U radu nisu korišteni programski alati kao što je *Simulink* (engl. *Simulation and Model-Based Design*) ili bilo koji drugi alati iz razloga prikaza jednostavnog prikupljanja podataka u stvarnom vremenu [3]. Na slici 1. prikazana je blok shema sustava za prikupljanje podataka sa senzora i prikaza istih uz pomoć Matlab-a.



Slika 1. Blok shema sustava za prikupljanje podataka sa senzora

Programski odsječak učitani na Arduino mikroupravljač u kojem se mjeri temperatura, vlažnost zraka i intenzitet svjetla prikazan je u nastavku, na slici 2.

Program za mjerenje temperature i vlažnosti (slika 2.) koristi gotovu knjižnicu naziva *DHTLib* koja sadržava funkcije za čitanje temperature i vlažnosti s AM2302 senzora [4]. Princip rada programa temelji se na mjerenju fizikalnih veličina i slanju izmjerenih rezultata preko komunikacijskog sučelja USB (engl. *Universal Serial Bus*) na računalo. Nakon inicijalizacije potrebnih parametara za rad senzora slijedi Arduino beskonačna petlja prikazana u nastavku (*void loop()*) kojom se pomoću funkcije *readHumidity()* čita vrijednost vlažnosti zraka, pa nakon toga pomoću funkcije *readTemperature()* vrijednost temperature. Intenzitet svjetla mjeri se uz pomoć funkcije *analogRead* koja se koristi za mjerenje veličina s analognih priključnica. Funkcija za mjerenje analognih veličina na izlazu vraća cjelobrojne vrijednosti u opsegu od 0 do 1023, gdje vrijednost 0 predstavlja 0V na izlazu iz senzora, a vrijednost 1023 predstavlja 5V na izlazu iz senzora što označuje maksimalni intenzitet svjetla, te se tako izmjerena vrijednost skalira uz pomoć funkcije *map* na vrijednosti od 0 do 100, što predstavlja vrijednosti intenziteta svjetla od 0 do 100%. Nakon mjerenja vrijednosti uz pomoć funkcije *Serial.println* vrijednosti se šalju na računalo. Funkcija *println* šalje vrijednost podataka kroz serijski port i nakon prikaza podatka prelazi u novi red u kojem ispisuje sljedeću vrijednost. Prema potrebi samog uzorkovanja, veličine se mogu učitavati u točno određenim vremenskim razmacima upotrebom *delay* funkcije na kraju petlje. Primjer prikazan na slici 3. pokazuje očitavanje vremenskog perioda od 1s.

Programski kod sa slike 2. izvodi se u mikroupravljaču i namijenjen je za mjerenje vlažnosti, temperature zraka i intenziteta svjetlosti. S druge strane Matlab programski kod izvodi se na računalo i namijenjen je prikupljanju podatak s komunikacijskog sučelja.

```
void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  int svjetla = map(analogRead(analogPin), 0, 1023, 0, 100);
  Serial.println(h);
  Serial.println(t);
  Serial.println(int_svjetla);
  delay(1000);
}
```

Slika 2. Odsječak programskog koda za mikroupravljač

Programski kod kojim Matlab komunicira s mikroupravljačem prikazan je u nastavku. Izvodi se kao skripta učitana u Matlab programski paket.

Program uspostavlja serijsku komunikaciju na određenom komunikacijskom sučelju, u ovom slučaju je to serijsko komunikacijsko sučelje (engl. *COM port*) s mikroupravljačem prema zadanoj brzini komunikacije koja treba biti ista od strane mikroupravljača i Matlab programskog paketa. Nakon otvaranja komunikacijskog sučelja slijedi *while* petlja prikazana u nastavku koja čita podatke s mikroupravljača. Podaci se čitaju uz pomoć

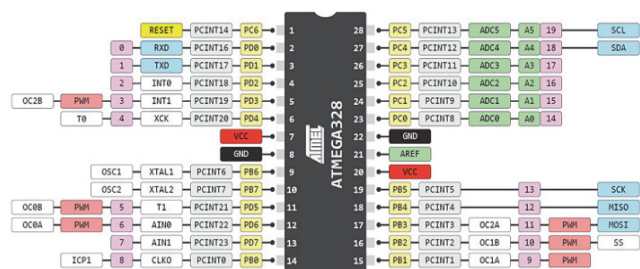
fscan funkcije i to svaka veličina posebno. Svaka se veličina prikazuje na posebnom grafu, a svaka nova vrijednosti se iscrtava odmah pri njezinom primitku. Nakon primitka unaprijed definiranih broja uzoraka (*interval = 500*) zatvara se komunikacijsko sučelje s mikroupravljačem. Ako je vrijeme čekanja do sljedećeg mjerenja na mikroupravljaču postavljeno na 1000ms (*delay(1000)*), a unaprijed definirani broj uzoraka u Matlab-u postavljen na 500, dobit ćemo 500 vrijednosti mjerenja temperature, vlažnosti i intenziteta svjetlosti svake sekunde.

```
while (i<interval)
a= fscanf (arduino, '%f' );
b= fscanf (arduino, '%f' );
c= fscanf (arduino, '%d' );
x=[x,a];
y=[y,b];
z=[z,c];
subplot(2,2,1);
plot(x);
title('Mjerenje vlažnosti u %');
xlabel('Vrijeme (s)');
ylabel('Izmjerena vrijednost u %');
axis([2, interval, 0, 100]);
grid on;
subplot(2,2,[3,4]);
plot(y, 'r');
title('Mjerenje temperature u °C');
xlabel('Vrijeme (s)');
ylabel('Izmjerena vrijednost u °C');
axis([0, interval, -40, 80]);
grid on;
subplot(2,2,2);
plot(z, 'g');
title('Mjerenje intenziteta
svjetlosti u %');
xlabel('Vrijeme (s)');
ylabel('Izmjerena vrijednost u %');
axis([0, interval, 0, 100]);
grid on;
i=i+p;
drawnow;
end
```

Slika 3. Odsječak programske skripte za Matlab

3. SKLOPOVSKI DIO SUSTAVA ZA MJERENJE FIZIKALNIH VELIČINA

Sklopovski dio ovog sustava dizajniran je uz pomoć senzora temperature i vlažnosti AM2302, LDR (engl. *Light Dependent Resistor*) senzora kojim se mjeri intenzitet svjetlosti i Arduino UNO R3 mikroupravljača. Temperaturni senzor radi u opsegu od -40 do +80°C s točnošću od +/- 0.5°C, senzor vlage radi u opsegu od 0 do 100% s odstupanjem od +/-2%.



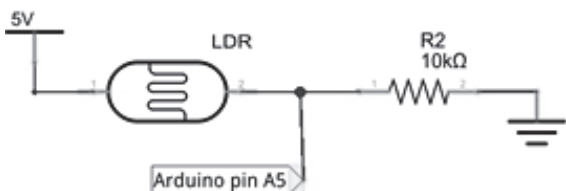
Slika 4. Popis priključnica mikroupravljača [5]

Na slici 4. prikazan je raspored priključnica ATmega328P mikroupravljača i raspored priključnica koje su povezane Arduino razvojnim okruženjem. Korišteni senzori spojeni su na priključnice prema tabeli:

Tabela 1. Raspored povezivanja senzora s mikroupravljačem

Senzor	Arduino	Atmega328P
LDR	Analog pin 5	Pin 28 – PC5
DHT22	Digital pin 2	Pin 4 – PD2

Povezivanje LDR senzora vrši se pomoću omskog dijelila kao što je prikazano na slici 5. LDR je jednostavan omski senzor kojem otpor na priključnicama ovisi o intenzitetu osvjetljenja. Pri spajanju na mikroupravljač promjena otpora LDR senzora rezultira promjenom napona na njegovim priključnicama. Promjena napona na priključnicama LDR senzora mjeri se uz pomoć 10 bitovnog A/D (analogno digitalnog) pretvornika s Arduino razvojnim okruženjem. Arduino na izlazu AD pretvorbe daje vrijednosti od 0 do 1023, gdje 0 predstavlja potpuni mrak, a 1023 maksimalno osvjetljenje (sunčevo zračenje iznad 10000 lux).



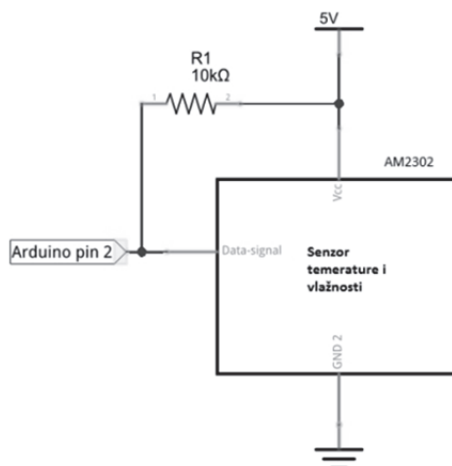
Slika 5. Shema spajanja LDR senzora na Arduino

Povezivanje senzora temperature i vlažnosti na Arduino prikazano je na slici 6. Za mjerenje je korišten senzor AM2302 koji se sastoji od kapacitivnog senzora za mjerenje vlažnosti i temperaturnog senzora visoke točnosti [4]. Senzor je tvornički kalibriran i u sebi sadrži mikroupravljač visokih performansi iz kojeg se pomoću komunikacije po jednoj žici (engl. *Single-bus communication*) čitaju informacije. Podatak o temperaturi ili vlažnosti sastoji se od 16 bita, dok ih mikroupravljač čita u blokovima od po 8 bita.

Komunikacija sa senzorom se odvija na sljedeći način: mikroupravljač šalje start signal senzoru (postavlja logičku '0' minimalno 800 μs na priključnicu s koje dohvaća podatke) te nakon toga AM2302 senzor prelazi sa stanja mirovanja (engl. *sleep*) u stanje brzog načina rada (engl. *high-speed*). Nakon što senzor pošalje signal odziva prema mikroupravljaču (prema standardu je to logička '0' 80 μs pa nakon toga logička '1' 80 μs) šalje se 40 bita u blokovima od 8 bita podataka sljedećim redoslijedom:

- 8 bitna vrijednost vlažnosti zraka prije decimalne točke,
- 8 bitna vrijednost vlažnosti zraka poslije decimalne točke,
- 8 bitna vrijednost temperature zraka prije decimalne točke
- 8 bitna vrijednost temperature zraka poslije decimalne točke,
- Paritetni bit.

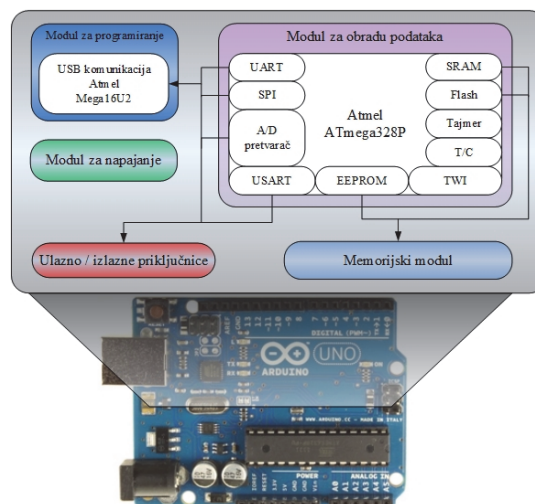
Nakon poslanih svih 40 bita senzor prelazi u stanje mirovanja do sljedećeg čitanja podataka [4].



Slika 6. Shema spajanja senzora temperature i vlage na Arduino

4. RAZVOJNO OKRUŽENJE ARDUINO

Razvojno okruženje Arduino je otvorenog koda (engl. *open source*) i otvorenog sklopovlja (engl. *open hardware*) razvojna okolina, temeljena na fleksibilnom i za uporabu jednostavnom sklopovlju te programskoj okolini sličnoj Javi ili C-u. Dizajniran je kako bi proces uporabe elektronike u multidisciplinarnim projektima bio jednostavniji. Omogućava jednostavno i povoljno sučelje za stvaranje kreativnih i korisnih projekata. S obzirom da je Arduino platforma otvorenog tipa, dozvoljeno je njezino dijeljenje i preuređivanje u svrhu kreiranja novih platforma koje su međusobno kompatibilne tako da su razvojem nastale još mnoge inačice razvojnih okruženja baziranih na Arduino platformi. U osnovi, sva Arduino kompatibilna sklopovlja sastoje se od mikroupravljača, integriranog sklopa za komunikaciju s računalom te perifernih elektroničkih dijelova za osiguranje mogućnosti rada mikroupravljača – stabilizatori napona, kvarcni oscilator za generiranje frekvencije takta i slično [6].



Slika 7. Arhitektura arduino sklopovlja

Razvojna okolina Arduino UNO R3 sadrži 14 digitalnih ulaza/izlaza, 6 analognih ulaza/izlaza i podržava USB prijenos podataka. Ima ugrađen mikroupravljač ATmega328P koji radi s naponom od 5V. Najveća dopuštena struja na ulazno/izlaznim priključnicama je 40 mA i svaka priključnica ima ugrađeni unutarnji pritezni otpornik (engl. *Pull-up resistor*) veličine 20 do 50 kΩ. Flash memorija je veličine 32 KB od čega je 0,5 KB rezervirano za program kojim se učitavaju vlastiti programi (engl. *bootloader*). Brzina procesora je 16 MHz, a količina radne memorije je 2 KB. Napaja se preko USB sučelja računala ili preko vanjskog napajanja gdje je najpogodnije vanjsko napajanje baterija od 9V.

Kroz digitalne izlaze korisnici mogu povezati različite elektroničke uređaje kao što su LED, senzori, zvučnici, motori i slično te ih zatim kontrolirati mikroupravljačem. Kao što je prikazano na slici 7, arhitektura sklopovlja Arduina uglavnom se sastoji od modula za napajanje, modula za programiranje, memorijskog modula i modula za obradu podataka.

Modul za napajanje je odgovoran za konstantan istosmjerni napon od 5V preko USB sučelja ili preko vanjskog adaptera gdje napon smije biti u granicama od 7 do 12 VDC. Modul za programiranje je odgovoran za prebacivanje programskog koda u mikroupravljač kroz komunikacijsko sučelje UART (engl. *Universal Asynchronous Receiver /Transmitter*) na računalo. Memorijski modul je odgovoran za upravljanje memorijama tipa EEPROM i SRAM. Modul za obradu podataka rabi Atmel ATmega328P mikroupravljač i odgovoran je za analizu i obradu podataka. On također kontrolira rad svake komponente i upravlja izlaznim naponom svake priključnice na Arduino razvojnoj okolini.

4.1. Digitalne priključnice Arduino Uno modela

Svaki digitalni priključak razvojnog sklopa Arduino označen brojevima od 0 do 13 može se rabiti kao ulaz ili izlaz. Maksimalni napon koji se može pojaviti na izlaznim digitalnim nožicama je 5V. Svaka digitalna priključnica pored ulaznih i izlaznih funkcija ima još i druge funkcije:

- **0 RX i 1 TX:** za primanje (RX) i slanje (TX) serijskih podataka. Te dvije priključnice su povezane s istoimenim priključnicama na ATmega8U2 ili ATmega16U2 mikroupravljačima i zaslužni su za USB komunikaciju.
- **2 i 3:** omogućavaju izlazni prekid (engl. *External interrupt*).
- **3, 5, 6, 9, 10 i 11:** omogućavaju pulsno širinsku modulaciju (engl. *Pulse Width Modulation ili PWM*) pri frekvenciji 490Hz.
- **10(SS), 11(MOSI), 12(MISO), 13(SCK):** priključnice omogućuju komunikaciju preko serijskog sinkronog modula (engl. *Serial Peripheral Interface ili SPI*).
- **13:** na priključnici je povezana LED (engl. *Light-Emitting Diode*). Ako je priključena na napajanje, LED svijetli. Ne koristi se kao digitalni ulaz zbog već spojene LED.

4.2. Analogne priključnice Arduino Uno modela

Na raspolaganju imamo 6 analognih ulazno/izlaznih priključnica s oznakama A0 do A5. Svaki ATmega mikroupravljač kojega rabe Arduino moduli ima 6 kanalni analogni digitalni pretvornik (engl. *Analog-to-digital converter*). Pretvornik je rezolucije 10 bita i vraća cjelobrojne vrijednosti od 0 do 1023, što odgovara ulaznim naponima od 0 do 3.3V ili 5V, ovisno o referentnom naponu. Glavna funkcija analognih ulaza za većinu Arduino modula je pretvorba analognih vrijednosti s različitih senzora, ali ih lako možemo rabiti i kao digitalne ulaze i izlaze koje zovemo GPIO (engl. *General Purpose Input/Output*). Za korištenje analogne priključnice kao digitalne odlučujemo se u načelu samo ako nam nedostaje digitalnih nožica na projektu. Ulazi također imaju ugrađeni pritezni otpornik koji možemo lako programski uključiti. Dodatnu funkcionalnost imaju priključnice A4 (SDA) i A5 (SCL) koje omogućuju TWI (engl. *Two Wire Interface*) komunikaciju.

4.3. Napajanje

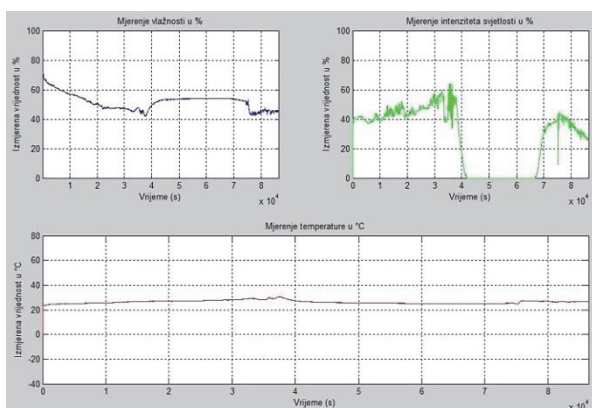
Priključnice za napajanje koriste se za dostavljanje električne energije modulima:

- **VIN** (engl. *Input Voltage*): upotrebljava se za napajanje Arduino modula dok se baterija rabi kao vanjsko napajanje. Možemo ga upotrebljavati i kao izlazni napon i napajati druge elemente sheme ako imamo modul priključen na napajanje preko adaptera.
- **GND:** dvije priključnice za referentnu točku, tj. zajedničku masu.
- **5V:** izlazni napon od 5V sa stabilizatora napona na modulu.
- **3.3V:** izlazni napon od 3.3V sa stabilizatora napona na modulu maksimalne struje 50mA.
- **RESET:** ako ovu priključnicu spojimo na masu, modul se resetira. Na priključnicu već je spojeno tipkalo koje se nalazi na samom modulu i rabimo ga za resetiranje cijelog modula.
- **IOREF** (samo UNO verzije 3): omogućuje Arduino modulu da se prilagodi naponu napajanja kojeg upotrebljava mikroupravljač na ulazno izlaznim priključnicama. Moguće je postaviti 3.3 ili 5V.

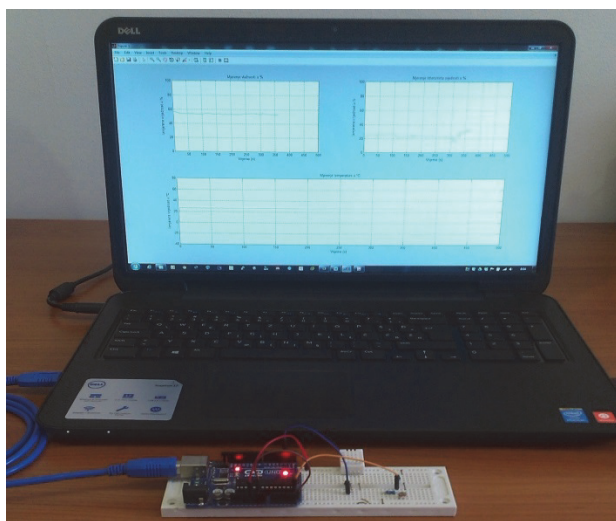
5. REZULTATI

Na slici 8. prikazani su rezultati mjerenja temperature, vlažnosti zraka i intenziteta svjetlosti. Graf temperature iscrtava se u unaprijed zadanom vremenu po osi ordinate, a po osi apscise ispisane su vrijednosti temperature. Temperatura os postavljena je od -40 °C do 80°C što predstavlja minimum i maksimum prema karakteristikama temperaturnog senzora. Svaka izmjerena veličina prikazana je na posebnom grafu u periodu od 24 sata. Os ordinate vlažnosti i intenziteta osvijetljenja prikazuju se u rasponu od 0% do 100%. Kroz prikazane rezultate u stvarnom vremenu možemo odrediti da li senzor radi dobro te uvidjeti postoje li

poteškoće koje se javljaju unutar nekog sustava u kojem se vrši mjerenje.



Slika 8. Rezultati 24 satnog mjerenja



Slika 9. Prikaz sklopa spojenog na eksperimentalnoj pločici

6. ZAKLJUČAK

Složenost prikupljanja podataka uz pomoć korištenja Matlab-a s Arduino razvojnim sklopom, znatno se smanjuje. Mogućnost primjene ovakve analize podataka proteže se od industrijskih postrojenja do mjerenja parametara sa senzora u kućanstvu. U radu je prikazano kako se Matlab može koristiti kao sučelje za bilo koji analogni uređaj ili senzor uz pomoć Arduino razvojne okoline te ujedno tako primiti i spremati podatke različitih fizikalnih veličina za buduću uporabu.

U nastavku ovog projekta moguće je spojiti više senzora i izvršnih elemenata kojima bi mogli automatizirati rad nekog sustava na temelju izmjerenih podataka. Također, postoji mogućnost spajanja nekih od RF komunikacijskih modula (npr. *Bluetooth*) kako bi izbjegli povezivanje sklopa s računalom žičanim putem.

7. LITERATURA

- [1] <http://www.mathworks.com>. (Dostupno: 25.7.2014.).
- [2] S. F. Barrett; D. J. Pack: *Microcontrollers Fundamentals for Engineers and Scientists*, Vol. 1, No. 1 (2006) 1–124.
- [3] <http://www.mathworks.com/products/simulink/> (Dostupno: 25.7.2014.).
- [4] <http://metebox.tk/files/AM2302.pdf>. (Dostupno: 25.7.2014.).
- [5] <http://arduino-info.wikispaces.com/QuickRef>. (Dostupno: 25.7.2014.).
- [6] <http://arduino.cc/en/Main/arduinoBoardUno>. (Dostupno: 25.7.2014.).
- [7] Borah, S.; Scholar, M. T.: *Temperature Monitoring of Server Room Using Matlab and Arduino*, IJERT, Vol. 2, No. 9, (2013)849–851
- [8] Lee, S.; Li, Y.; Kapila, V.: *Development of a Matlab-Based Graphical User Interface Environment for PIC Microcontroller Projects*, 2004.
- [9] Potdar, R. M.; Pathak, P.: *Microcontroller based real time data acquisition system with parameter analyzer using Matlab*, IJERIA, Vol. 2, No. 5, (2009) 183-195

Kontakt autora:

Bruno Palašek, struč.spec.ing.el.
 Međimursko veleučilište u Čakovcu
 BANA JOSIPA JELAČIĆA 22a, Čakovec
 bruno.palasek@mev.hr

Petra Mesarić, mag.ing.el.techn.inf.
 Međimursko veleučilište u Čakovcu
 BANA JOSIPA JELAČIĆA 22a, Čakovec
 petra.mesaric@mev.hr

dr.sc. Mihael Kukec, viši predavač
 Međimursko veleučilište u Čakovcu
 BANA JOSIPA JELAČIĆA 22a, Čakovec
 mihael.kukec@mev.hr