

Sveučilište u Zagrebu  
Fakultet strojarstva i brodogradnje

Igor Bolteković Tkalec  
Udaljeni nadzor vibracijskog stanja stroja

Zagreb, 2013. godina

Ovaj rad izrađen je na Fakultetu strojarstva i brodogradnje - Zavod za industrijsko inženjerstvo, Laboratorij za operacijska istraživanja, pod vodstvom dr.sc. Dragutina Lisjaka i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2012./2013.

## Popis i objašnjenja kratica i oznaka

<b>LAN</b>	<i>engl. Local area network – lokalna računalna mreža</i>
<b>WAN</b>	<i>engl. Wide area network – širokopojasna računalna mreža</i>
<b>smt</b>	<i>engl. surface-mount-technology – tehnologija površinskih elektrokomponentata</i>
<b>MEMS</b>	<i>engl. <b>Microelectromechanical systems-</b></i>
<b>RXD</b>	<i>engl. Recieve dana – primljeni podaci</i>
<b>TXD</b>	<i>engl. Trensmitted dana-odaslani podaci</i>
<b>PWM</b>	<i>engl. <b>Pulse-width modulation – pulsno- šinska modulacija</b></i>
<b>USB</b>	<i>engl. Universal serial bus</i>
<b>RS-232</b>	<i>oznaka za standardnu serijsku komunikaciju</i>
<b>SSH</b>	<i>engl. Secure shell- siguran pristup udaljenom shellu</i>
<b>Shell</b>	<i>-korisničko, tekstualno sučelje UNIX (i *NIX) sustava</i>
<b>Telnet</b>	<i>mrežni protokol za udaljenu, virtualnu terminalnu komunikaciju</i>
<b>PHP</b>	<i>engl. <b>Hypertext Preprocessor, serverski programski jezik</b></i>
<b>HTML</b>	<i>engl. HyperText Markup Language- programski jezik web stranica</i>
<b>JS(JavaScript)</b>	<i>oznaka za skriptni jezik, sa strane klijentovog pretraživača</i>

## Sadržaj

1. Uvod .....	2
2. Opći i specifični ciljevi rada .....	3
3. Plan rada .....	5
4. Materijali i metode .....	6
4.1. Akcelerometar .....	6
4.2. Mikrokontroler .....	10
4.3. Router .....	12
4.4. Fizički prototip sustava – objedinjenje funkcionalnih elemenata .....	15
5. Rezultati .....	20
6. Rasprava .....	23
7. Zaključci .....	24
Zahvale .....	25
Popis literature .....	26
Sažetak na hrvatskom jeziku .....	27
Summary .....	28
Životopis .....	29
PRILOG 1 .....	30
Prilog 2 .....	41

## **1. Uvod**

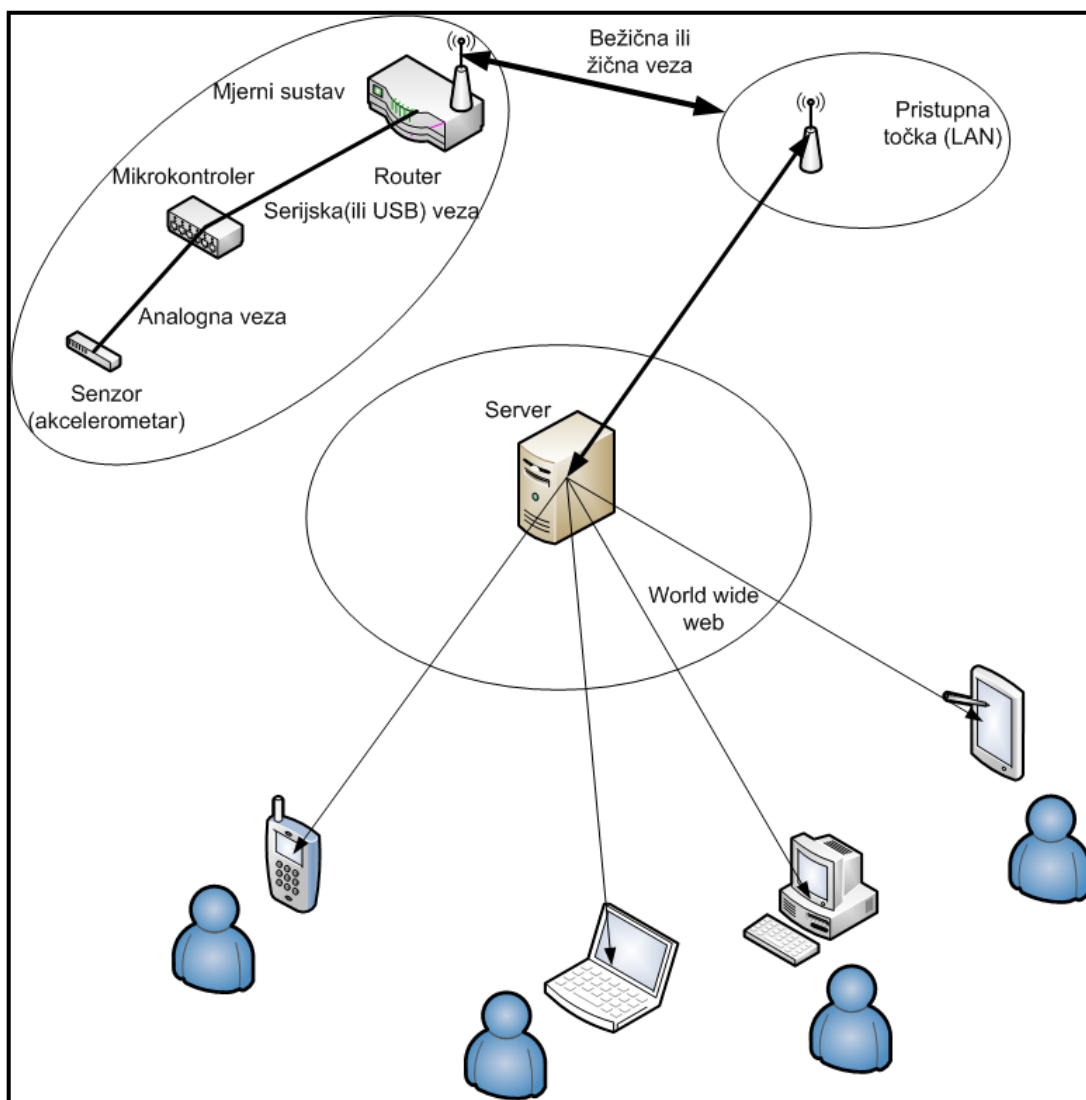
U današnje vrijeme progresivnog razvoja tehnologija, ponekad izostaje nužna komponenta objedinjavanja istih radi boljitka zajednice, bilo kao ekonomske strukture, bilo kao društvene cjeline. Kako živimo u društvu orijentiranom na stvaranje vrijednosti, svako poboljšanje tehnologija stvaranja vrijednosti (njihovog optimiranja) daje pozitivne rezultate i u ekonomskom smislu, što neminovno vodi ka poboljšanju društva u cjelini. Primjere navedene teze možemo pronaći i u bližoj i daljoj prošlosti, pa primjerice pronalazak pluga u poljodjelstvu značio je ne samo lakši fizički rad za poljodjelce, već i širi društveni značaj, jer se je na istoj jedinici površine, sa jednakim brojem radne snage moglo proizvoditi više hrane, sniziti cijenu iste, te većem broju ljudi omogućiti egzistenciju. Danas smo svjesni izuzetnog razvoja tehnologija i njihovih rezultata na gotovo svim poljima ljudske djelatnosti, ali ponekad se nađemo u situaciji da tehnologije, ma kako one zasebno bile razvijene, često nisu integrirane u zajedničku cjelinu, iako se tome sve više teži, i sve više upravo integracija tehnologija postaje važnija od samog razvoja istih.

Nadalje, u smislu eksploatacije tehnologija u realnom okruženju, neminovno slijedi trenutak otkazivanja funkcionalnosti takvog sustava, te je stoga razvijena tehnička disciplina održavanja, koja za glavi cilj svojeg djelovanja ima upravo spriječiti nenadani zastoj funkcionalnosti, to jest održavati sustav u zadanim parametrima djelovanja, služeći se dostupnim metodama i tehnologijama. Za preventivno održavanje izuzetno su bitne tehnologije nadzora i definiranja trenutnog stanja sustava, kako bi se čim preciznije moglo odrediti i buduće stanje, to jest trenutak u kojem će sustav otkazati. Jedna od takvih metoda jest i mjerenje vibracijskog stanja sustava (konkretno uređaja ili stroja). Uređaji za mjerenje vibracija danas su izuzetno razvijeni, ali je njihova upotreba manualna, što znači da kod svakog očitavanja vibracijskog stanja, operater fizički mora obaviti mjerenje na željenom mjestu sustava, te tako prikupljene podatke kasnije analizirati kako bi se došlo do saznanja o stanju sustava, te eventualnim mjerama ukoliko parametri pokazuju da je sustav u kritičnom stanju.

Ovaj rad za cilj ima upravo povezivanje višestruko dostupnih tehnologija, sa svrhom omogućavanja udaljenog nadzora vibracijskog stanja mjerenog sustava, a sve zbog efikasnijeg održavanja tog sustava i sprečavanja nenadanih, neželjenih posljedica u vidu prekida funkcionalnosti sustava, što se uvijek odražava zastojem osnovne djelatnosti u proizvodnom sustavu, što dovodi do ekonomskih troškova, koji su se boljim nadzorom mogli spriječiti.

## 2. Opći i specifični ciljevi rada

Cilj ovog rada jest osmisлити i fizički ostvariti sustav koji će od udaljenog mjesta mjerenja vibracijskog stanja, korisniku omogućiti platformu za pristup rezultatima mjerenja, sa činjenicom da je udaljeni sustav (uređaj) samostalan te da se izuzme potreba za računalom korisnika kao mjesta prikupljanja podataka, već da korisnik pomoću bilo kojih danas dostupnih uređaja za mrežnu komunikaciju (bilo stolno računalo, prijenosnik, tablet računalo ili *smartphone*) pristupi željenim rezultatima mjerenja. Udaljeni sustav za prikupljanje i pripremu podataka, prema ovako postavljenom cilju, na mjernom mjestu mora imati osiguran samo izvor napajanja (putem standardne naponske mreže), te pristup internetu, bilo putem lokalne žične mreže (LAN sa prisupom globalnoj mreži), ili *wireless* pristupnoj točki. Ovakav cilj podrazumijeva objedinjavanje sustava za prikupljanje podataka (senzor, mikrokontroler), te mrežne komponente (*routera*) koji pak za zadatak ima proslijediti podatke na udaljeno računalo sustava (*server*).



Slika 1- Shematski prikaz modela sustava

Dakle, deklarirani ciljevi rada su:

**Opći cilj rada:** osmisliti model udaljenog nadzora vibracijskog stanja mjenog sustava, te ostvariti fizički prototip takvog modela

**Konačni specifični cilj:** omogućiti pristup očitanjima senzora akcelerometra putem Internet preglednika, neovisno o platformi korisničkog sustava, u grafičkom načinu te tekstualnom obliku pogodnom za daljnju analizu, na bilo koji od postojećih načina

Iz shematskog prikaza modela sustava jasno je da su uređaji objedinjeni kao „Mjerni sustav“ glavni fokus ovog rada, te jest želja postići modularnost, i sa strane ugrađivanja uređaja, i sa strane korisnika i njegovog načina pristupa sustavu. Upravo jest integracija i modularnost postojećih tehnologija u svrhu stvaranja novih mogućnosti njihova korištenja i tema ovog rada i trend u tehnici općenito. Prednosti sustava putem globalne mreže računala jest višestruka, a obuhvaća pristupačnost neovisno o platformi na kojoj se koristi, te kao takav standard jedan jest od rijetkih koji bez preinaka radi na svim računalnim platformama. Ova činjenica izuzima korisnika iz domene brige oko programske podrške i fokusira se samo na njegovo korištenje.

### **3. Plan rada**

Prema sljedećim navodima ostvaren je ovaj rad, te njegov rezultat funkcionalni sustav, a svaki njegov element u gotovoj izvedbi prikazan je u poglavlju „Materijali i metode“:

Plan rada:

- izrada modela sustava nadzora
- definiranje fizičkog ostvarenja
- definiranje toka podataka u sustavu
- odabir najpogodnijih tehnologija, to jest programskih jezika, za svaku komponentu sustava
- izrada fizičkog prototipa, objedinjavanje elektrotehničkih komponenti, prema prethodno definiranom modelu sustava
- izrada programske podrške prema zamišljenom modelu, a za fizički prototip:
- izrada programa mikrokontrolera dužnog za prikupljanje podataka sa senzora (akcelerometra) i slanje prema mrežnoj komponenti (router)
- podešavanje mrežne komponente sustava(router) za prihvaćanje podataka sa mikrokontrolera te njegovog prosljeđivanja na server
- izrada serverske skripte za prihvatanje podataka sa mrežne komponente sustava (router) te priprema, obrada i spremanje podataka u odgovarajućem obliku za svrhu izuzimanja kod zahtjeva korisnika (to jest korisničkog Internet preglednika)
- izrada korisničkog web sučelja (web stranice), sa osnovnom svrhom izuzimanja prethodno pripremljenih podataka na serveru, te prikazivanja istih, u tekstualnom i grafičkom obliku



## 4. Materijali i metode

Prema definiranom planu rada, proizašlom modelu sustava i fizičkom modelu sustava materijali koje su korišteni u radu su sljedeći:

-senzor – akcelerometar;

-mikrokontroler;

-router<sup>1</sup>;

-server;

-korisnički uređaj;

Svaki od navedenih materijala ujedno predstavlja i funkcionalan element cijelog sustava, te će sustav funkcionirati jedino ako svi navedeni elementi zasebno funkcioniraju.

### 4.1. Akcelerometar

Akcelerometar jest vrsta senzora, koji ima sposobnost bilježiti promjenu ubrzanja po osi. Danas su izuzetno popularni u korisničkim uređajima široke potrošnje (tablet računala, mobilni *smart* telefoni, igrače konzole), zbog jednostavnosti izrade u obliku integriranog kruga (*smt chipa*), izrađenih MEMS tehnologijama. Ovisno o izvedbi postoje akcelerometri koji mjere ubrzanja u jednoj, dvije ili sve tri osi. Za potrebe odabranog modela, u konkretnom sustavu se koristi akcelerometar MMA 7361, koji ima sposobnost bilježenja ubrzanja u sve tri osi. Detaljne karakteristike dane su sljedećom listom:

-radni napon : 2,2 V do 3,6 V

-osjetljivost: 800mV/g, na 1,5g režimu, radnoj temperaturi od 25°C, i radnom naponu 3,3V

-dva režima:  $\pm 1.5g$ ,  $\pm 6g$

-sposobnost detekcije slobodnog pada

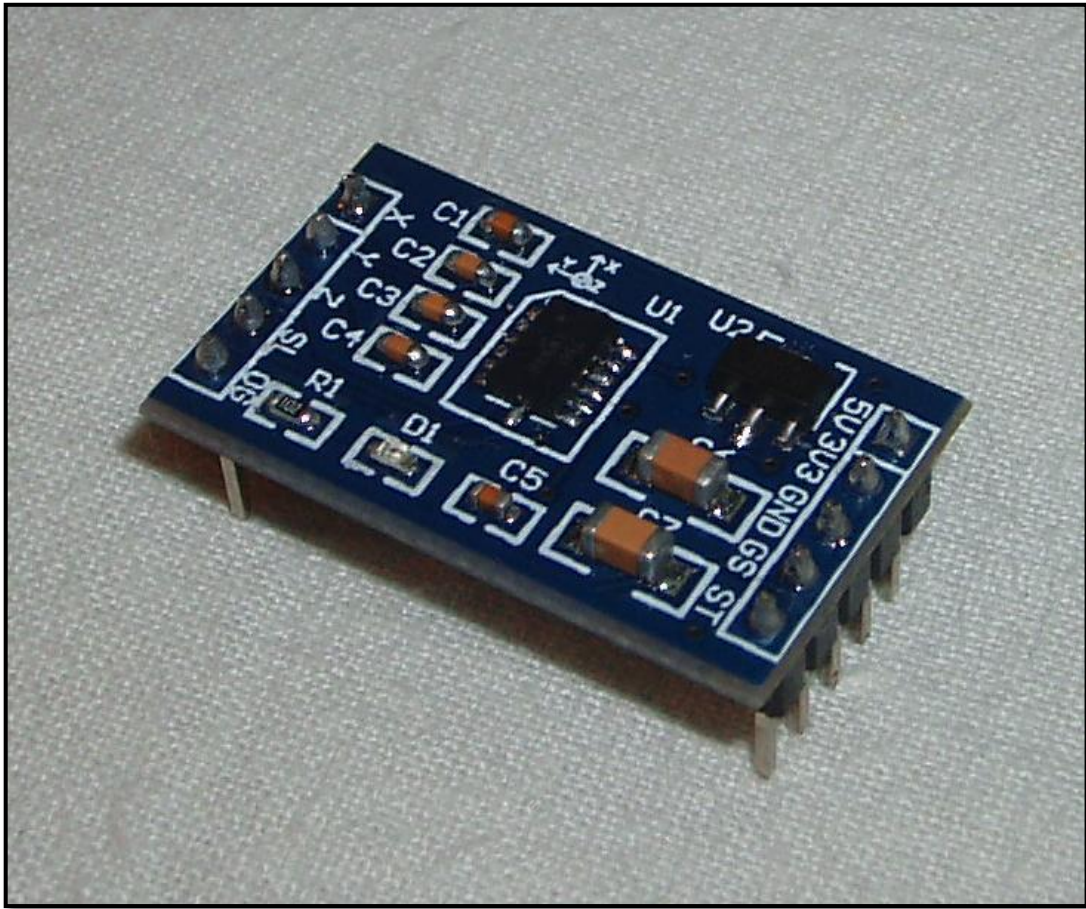
-*low pass* filter, kompenzacija temperature

-unutarnja frekvencija *sampliranja* (prikupljanja podataka) jest 11kHz

---

<sup>1</sup> Zbog nedostatka hrvatske inačice riječi *router* koristi se engleski naziv, detaljno pojašnjenje značenja same riječi navedena je pod točkom 4.3. ovog rada

Navedeni akcelerometar u sustav je ugrađen u obliku integrirane pločice, kako bi se olakšao pristup izvodima, upravo u ovakve eksperimentalne svrhe. Konkretni akcelerometar prikazan je na sljedećoj slici, sa napomenom da je pločica dimenzija 28x17mm.



**Slika 2-Akcelerometar MMA7361**

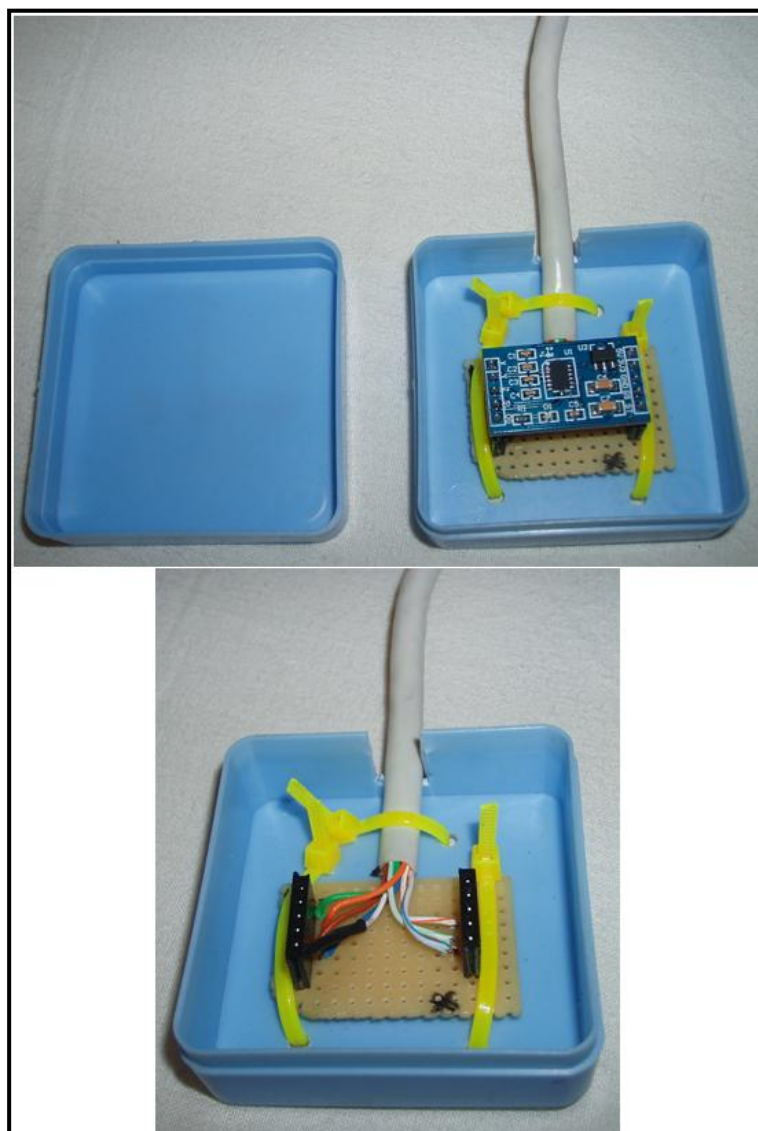
Navedeni akcelerometar ima analogne izvode, za očitavanja po svakoj osi. Predstavlja analogni uređaj, te je za čitanje podataka koje odašilje potreban analogni pretvornik u digitalni signal. Tu funkciju u ovom sustavu preuzima integrirani A/D pretvornik u mikrokontroleru, koji ima sposobnost rezolucije od 1024 segmenta. Pošto akcelerometar radi na naponu od 3.3V, to bi značilo da svaka promjena od minimalne vrijednosti jednog cijelog broja (0 -1023-Poglavlje 4.2) na strani mikrokontrolera, znači promjenu od 3,222 mV od vrijednosti koju šalje akcelerometar., te je ta vrijednost (od 3,222mV) minimalna vrijednost koja se treba promijeniti da bi ju A/D pretvornik registrirao.

Osjetljivost očitavanja senzora (kojeg šalje na izlaznim pinovima označenima sa X,Y,Z) jest 800mV/g pri režimu od  $\pm 1.5g$  (kojeg koristimo, zbog veće osjetljivosti nego pri režimu  $\pm 6g$ , gdje ona pada na 206mV/g). Navedena osjetljivost konkretno znači da se pri promjeni akceleracije za 1g, napon na izlazima promijeni za 800mV. Pošto mi najviše možemo registrirati promjenu od 3,22mV, to znači da je maksimalna osjetljivost koju registrira

mikrokontroler  $800/3,22=248,3\text{mV}$ , a u konačnici to iznosi  $0,004\text{g}$ . To je minimalna promjena koju sustav može registrirati, u idealnim uvjetima.

Problem koji se javlja zbog brzine rada A/D pretvornika ( $\sim 2\text{ms}$ ) jest ta da je maksimalna frekvencija koju može uređaj registrirati jednak  $500\text{Hz}$ , u idealnim uvjetima, ali je zbog prirode ovog rada (a ta je uspostavljanje veze između krajnjeg korisnika i mjernog uređaja), ta činjenica manje bitna. Sustav je postavljen tako da su svi elementi sustava funkcionalno u vezi, ali ujedno i modularni, što znači mogućnost zamjene bilo kojeg od elemenata, ako ispunjava barem one karakteristike koje trenutni element posjeduje.

Na narednoj slici (Slika 3.) prikazan je prihvat pločice akcelerometra, zajedno sa kućištem, te pripadajući kabel kojim je povezan senzor sa mikrokontrolerom. Ovakav postav je odabran kako bi se postigla fleksibilnost kod mjerenja vibracija, to jest fizički su odvojeni senzor i mikrokontroler.



**Slika 3- Kućište i prihvat akcelerometra**

Sljedeća tablica daje pregled oznaka i značenja svakog od izlaza, te opis spajanja istih na ostale elemente sustava.

Oznaka priključnog <i>pin-a</i>	Značenje	Napomena
ST	<i>Self-test</i>	NC
GS	<i>G-sensitivity</i>	Low- 1.5g, High 6g
GND	<i>Ground</i>	Uzemljenje
3V3	Priključak za 3.3V	
5V	Priključak za 3.3V	
X	Analogni izlaz očitavanja x osi	
Y	Analogni izlaz očitavanja y osi	
Z	Analogni izlaz očitavanja z osi	
SL	<i>Sleep</i>	High
0G	<i>Freefall detection</i>	NC

\*Napomena: oznake u tablici:

NC- *not connected* – nije nužno spajanje

Low – napon mora biti manji od 0.8V

High – napon mora biti veći od 1V, ne veći od 3.3V

Prema navodima u tablici konstruiran je prihvat pločice akcelerometra, pomoću kojeg se kablom i DB9 priključkom spaja na odgovarajuće ulaze pločice mikrokontrolera, kako je prikazano u poglavlju 4.4.

## 4.2. Mikrokontroler

„Mikrokontroler je derivat mikroprocesora, opremljenog dodatnim sklopovima koji mu omogućuju interakciju s okolišem. Mikrokontroler uzima informacije kao mjerne veličine, obrađuje ih po zadanom algoritmu i potom djeluje prema okolišu. Taj proces se odvija u tzv. „realnom vremenu“, tj. „on line“ [2].

Navedena definicija mikrokontrolera u suštini daje odgovor na pitanje zbog čega je on korišten u sustavu mjerenja. Naime, mikrokontroler posjeduje zapravo sve osnovne karakteristike kao i računalo, s prednostima malih dimenzija, male potrošnje energije, modularne interakcije sa okolinom (naime svaki mikrokontroler posjeduje I/O pinove, pomoću kojih je moguće upravljati periferijom na željeni način).

Mikrokontroler koji se koristi u radu jest *Arduino UNO R3*. Odabran je upravo ovaj mikrokontroler jer je njegova upotreba relativno jednostavna, moguće ga je programirati putem njegovog USB sučelja, a pomoću prilagođenog korisničkog sučelja relativno je jednostavno isprogramirati njegov zadatak. Prikazan je na sljedećoj slici.



Slika 4-Mikrokontroler Arduino UNO R3

Najvažnije karakteristike spomenutog uređaja dane su u sljedećem popisu:

- baziran je na ATmega328p mikrokontrolru (proizvođač *ATMEL*)
- ima 14 digitalnih I/O pinova, od kojih se 6 može koristiti kao PWM izlazi
- rezonira pomoću 16 Mhz keramičkog oscilatora
- mogućnost napajanja pomoću USB kabela, ili standardnog DC 9V, istosmjernog izvora
- radni napon 5V
- Flash memorija 32KB; SRAM-2KB, EEPROM 1KB
- Moguća UART TTL (5V) serijska komunikacija koja je dostupna na RX i TX pinova, sa indikatorskim LED svjetlima za nadzor komunikacije (u oba smjera)

Gornji navodi opisuju uređaj ali i uvode neka ograničenja i zakonitosti kod izrade sustava. Naime sklop radi na 5V, dok senzor treba pogoniti sa 3.3V. Na pločici mikrokontrolera dostupan je izvor od 3.3V, što bi bilo dovoljno za napajanje senzora, ali je potrebno još ulaz SL na pločici senzora podići na logički *HIGH* napon, to jest i na njemu osigurati napajanje od 3.3V. Iz tog razloga je na pločicu koja predstavlja vezu između senzora i mikrokontrolera ugrađen djelitelj napona, koji iz 5V osigurava potrebnih 3.3V za napajanje senzora.

Nadalje, upravo činjenica da je serijska komunikacija moguća i preko zasebnih pinova na pločici mikrokontrolera, u konačnici je omogućila povezivanje mrežnog uređaja (*routera*) i mikrokontrolera, jer operativni sustav *routera* ne podržava integrirani krug koji standardno služi za programiranje mikrokontrolera, ali podržava neke druge uređaje koji pretvaraju serijsku u USB komunikaciju. Konkretni uređaj korišten u sustavu opisan je u narednom poglavlju (integrirani krug pl2303).

### 4.3. Router

*Router* je mrežni uređaj koji omogućava povezivanje više mreža zajedno. Ima sposobnost stvoriti vlastitu mrežu, i tako korisnicima omogućiti međusobnu izmjenu podataka, ali istovremeno može povezati vlastitu (LAN) mrežu sa WAN, širom mrežom. Značajno se koristi u kućnim mrežama kao veza lokalne mreže sa svjetskom mrežom računala.

Pobliže gledano, *router* je zapravo računalo bez klasičnog korisničkog sučelja (periferija poput miša, tipkovnice i monitora), ali su svi ostali dijelovi koji čine klasično računalo prisutni. Upravo je ovakav uređaj odabran kao mrežna komponenta jer je široko dostupan, a pojedini modeli su i modularni, te se nad njima može steći potpuna kontrola.

Takav je model i u sustavu konkretno odabrani *Siemens sx763*. Više je razloga koji su utjecali da upravo ovaj model bude dio sustava. Naime, zahvatom na promjeni osnovnog *firmwarea* koji dolazi predinstaliran od strane nacionalnog operatera, može se postaviti u potpunosti otvoren operativni sustav, konkretno *OpenWRT*. Spomenuti je sustav zapravo ogoljena inačica linux operativnog sustava, a prilagođena za namjene na *routerima*. To činjenica je i glavni razlog odabira ovog modela, uz ostale prednosti široke pristupačnosti na tržištu, relativno niske cijene, ali i ne zanemarive (to jest iznimno značajne) činjenice da posjeduje USB port, koji nam je bitan zbog povezivanja sa prethodno spomenutim mikrokontrolerom, a putem sučelja integriranog kruga pl2303. Daljnje prednosti su mogućnost povezivanja na bilo koju dostupnu mrežu, bilo preko žice, ali i bežično. Naime, uređaj se može konfigurirati tako da bude klijent na mreži kao i svako drugo računalo, što sustavu za mjerenje vibracija pruža fleksibilnost u vidu samog mjesta očitavanja vibracijskog stanja, jer je za rad sustava u konačnici (u ovakvoj konfiguraciji) potreban samo izvor napajanja (standardno naponsko mjesto od 220V izmjenične struje na 50Hz), te pristupno mrežno mjesto.



Slika 5-Router Siemens sx763

Pošto router nema klasično korisničko sučelje (tipkovnica, miš monitor), kako je već spomenuto, interakcija s takvim uređajem moguća je na više načina. Gotovo svakom *router* u može se pristupiti serijskom vezom (nasljeđe još od vremena modema), telnet ili sigurnijom SSH konekcijom. Uobičajeni način jest upravo posljednje navedeni, te je to ujedno i način spajanja u toku ovog projekta.

Koraci provedeni u svrhu osposobljavanja routera za rad kako mu je namijenjeno modelom sustava uključivali su instalaciju novog operativnog sustava, (*open source*<sup>2</sup>*OpenWRT*), instalaciju potrebne podrške za konkretni slučaj (*driveri* za podršku USB konekcije, pl2303 integriranog kruga), te podešavanje mreže po modelu.

U radu *router* ima svrhu primanja podataka na serijskom portu, i slanja na udaljeni server putem TCP konekcije.

Osnovne karakteristike odabranog modela:

Brzina procesora: 333MHz

Veličina *flash* memorije: 8MiB

RAM: 32MiB

Sučelja: wireless, ethernet (x4), USB, Serisko, JTAG

---

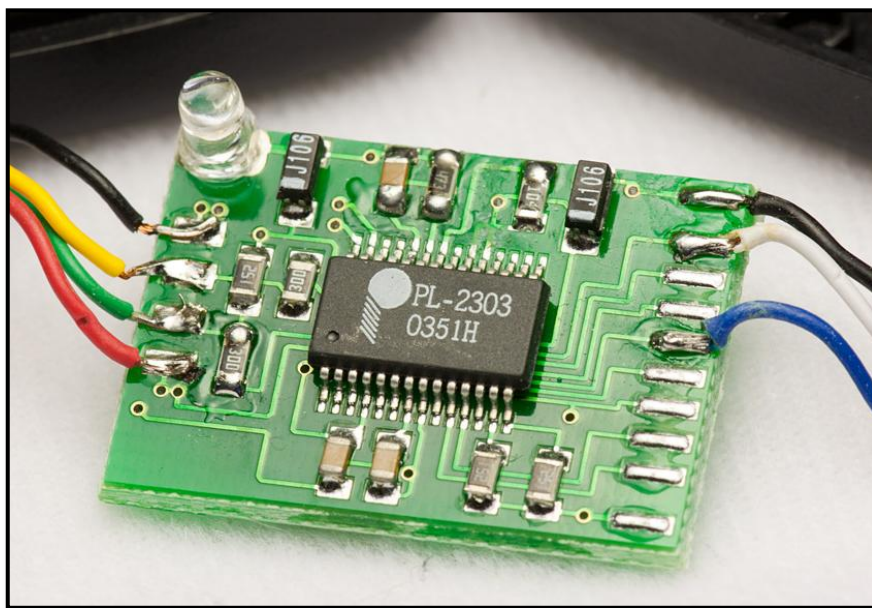
<sup>2</sup> Oznaka za programsku podršku nastalu u zajednici ljudi koji samo na dobrovoljnoj bazi razvijaju programska rješenja, na gotovo svim poljima danjašnje tehnologije, a svoja saznanja slobodno dijele u javnost



### Integrirani krug PL2303

Integrirani krug (*chip*) naziva pl2303 jest uređaj za pretvaranje serijskog digitalnog signala (RS-232) u moderniji USB signal. Ovaj je krug neizostavan dio sustava, jer omogućava povezivanje digitalnog serijskog izlaza iz mikrokontrolera (*RXD i TXD*) u USB sučelje mrežnog uređaja, to jest USB port *routera*. Bez ove komponente ne bi bilo moguće podatke prikupljene putem akcelerometra odaslati na server. Upravo ovaj sklop odabran je iz razloga što operacijsku sustav *routera* njega podržava, to jest za takvu vrst uređaja postoji potrebna programska podrška u vidu *drivera*. Kod prve iteracije izrade fizičkog prototipa nije uzet u obzir, jer se mislilo da USB sučelje samog mikrokontrolera može komunicirati sa mrežnim uređajem, koji bi za taj dio komunikacije imao FDTI integrirani sklop, ali, prema detaljnijoj analizi mikrokontrolera došlo je do saznanja da se komunikacija vrši ipak putem nekog drugog sklopa, točnije AVR-ovim (drugim) mikrokontrolerom koji oponaša navedeni FDTI *chip*. Iz tog razloga, direktna komunikacija mikrokontrolera putem njegovog USB sučelja i USB sučelja *routera* nije bila moguća, iako bi to bilo elegantnije rješenje.

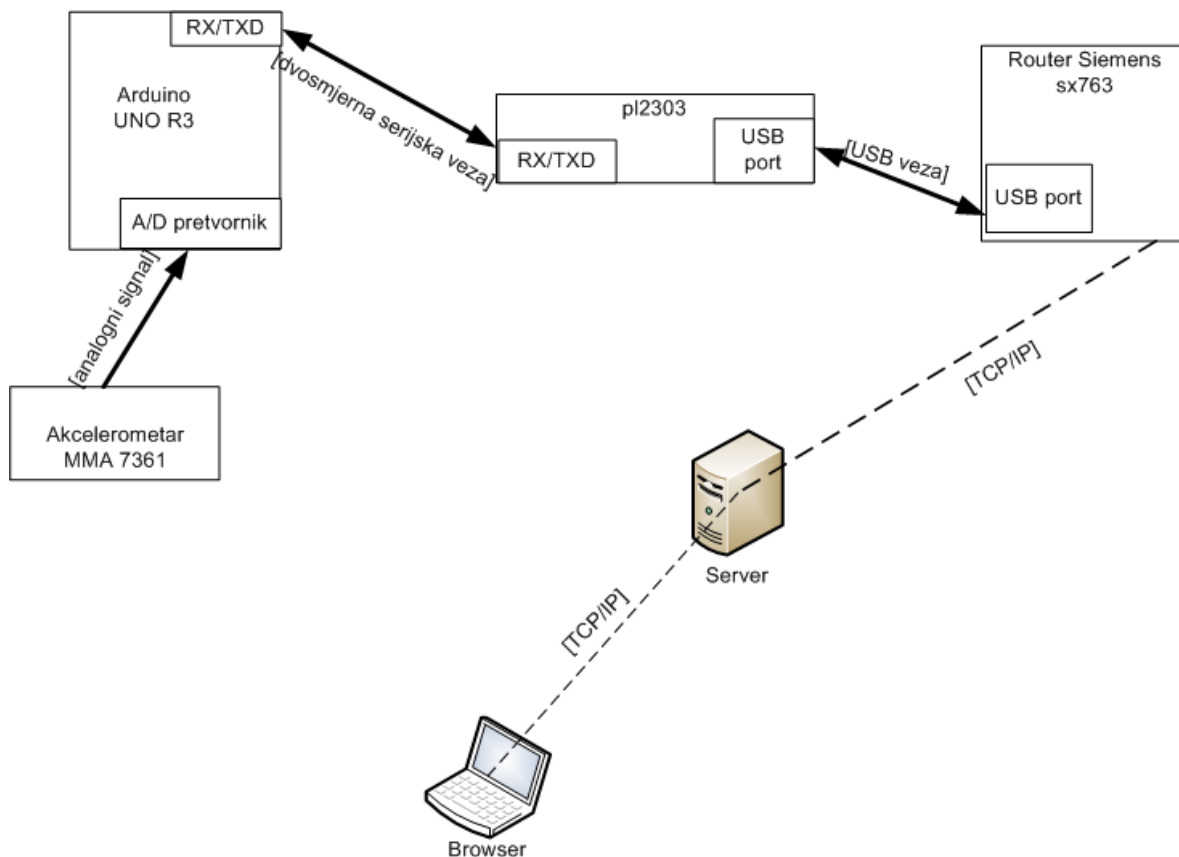
Posebne postavke navedenog sklopa nisu potrebne, on dolazi u obliku USB kabela sa standardnim priključkom s jedne strane, te, posebno za ovu namjenu prilagođenim sučeljem s druge strane, što je vidljivo na slici fizičkog prototipa, u poglavlju 4.3., te na sljedećoj slici



Slika 6-pl2303 integrirani krug

#### 4.4. Fizički prototip sustava – objedinjenje funkcionalnih elemenata

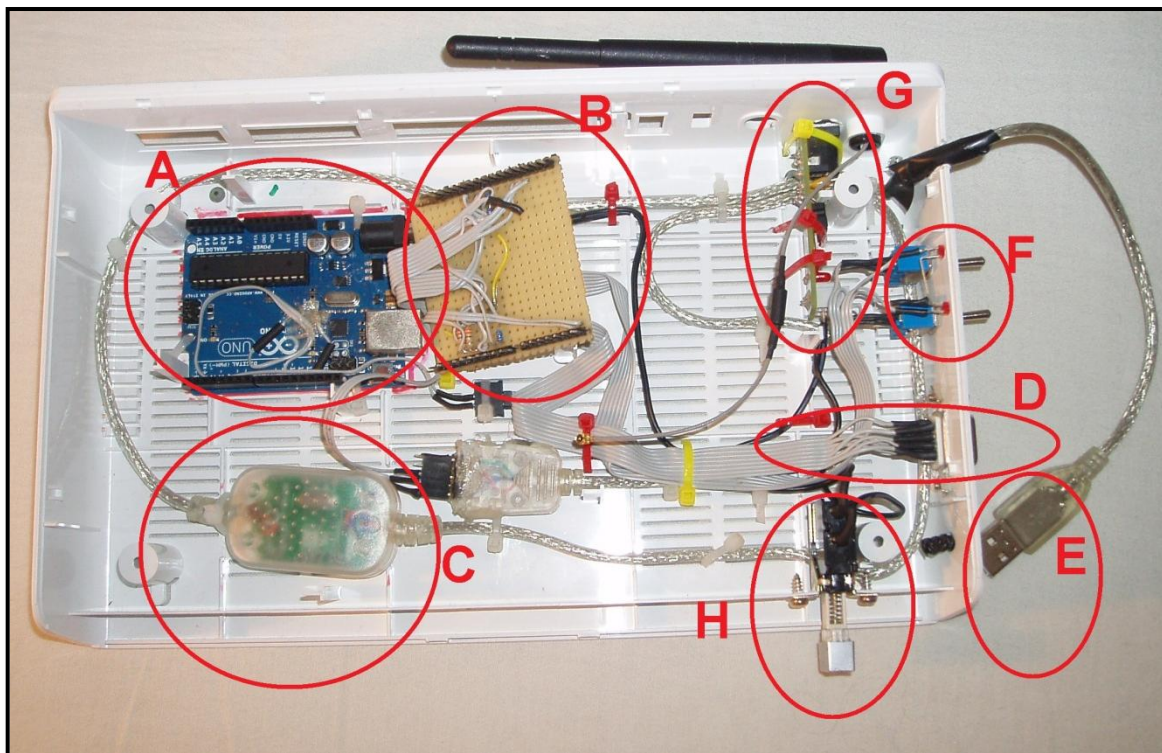
U točki 4. Navedeni su svi funkcionalni elementi sustava, a u ovoj točki oni se napokon predstavljaju objedinjeno, u svrhu ostvarivanja cilja rada. Slika 4. Prikazuje model fizičkog prototipa, te vrstu veza u sustavu.



Slika 7-Model fizičkog prototipa

Sa podatkovne strane najvažnije jest ostvariti komunikaciju između svih elemenata. Akcelerometar pruža analogni signal, mikrokontroler ga obrađuje u A/D pretvorniku, te odgovarajućim programskim kodom pretvara u serijski (RS-232)signal, koji pak putem pi2303 kabela dolazi do *router-a* koji putem mreže (bilo žično ili bežično) podatke šalje na port, na serveru na kojem se nalazi skripta (u php programskom jeziku), koja je prethodno otvorila *socket* („mjesto za spajanje“) i čeka na pridošle podatke. Spomenuta skripta ima za zadatak i prikupiti podatke te ih spremi u dvije datoteke, jednu koja služi kao *buffer* (za pohranu trenutnih podataka), te dnevnik (*log*) u koji se spremaju svi primljeni podaci od početka mjerenja. Trenutni podaci služe kao mjesto iz kojeg se u pregledniku prikupljaju podaci za iscrtavanje na grafu, a to se prikupljanje i osvježivanje, obavlja svakih 300 ms, što se pokazalo dovoljnim, jer za to vrijeme php skripta stigne pripremiti novi set podataka, a s druge strane, preglednik je sposoban iscrtati te podatke u navedenom vremenu.

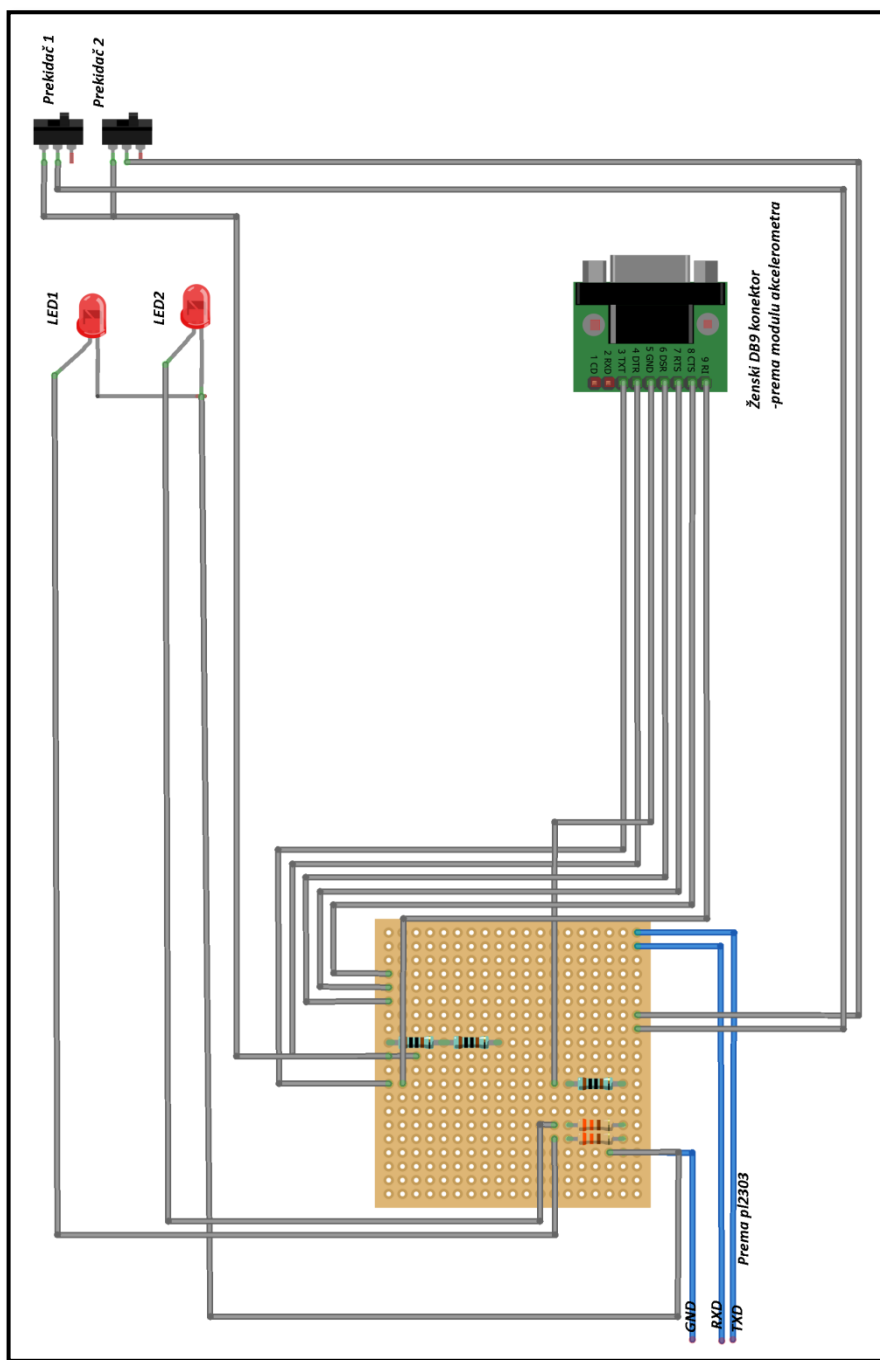
Zbog fizički dovoljnog mjesta u kućištu odabranog routera, odlučeno je da se sve komponente sustava smjeste upravo unutar kućišta, kao bi se postigla veća razina fleksibilnosti, te kako bi se umanjila mogućnost neželjene interakcije sa elementima sustava, to jest kako bi sustav bio čim je moguće više zaštićen od vanjskih utjecaja. Sljedeća slika prikazuje kako je sustav smješten unutar kućišta



**Slika 8-Fotografija fizičkog prototipa**

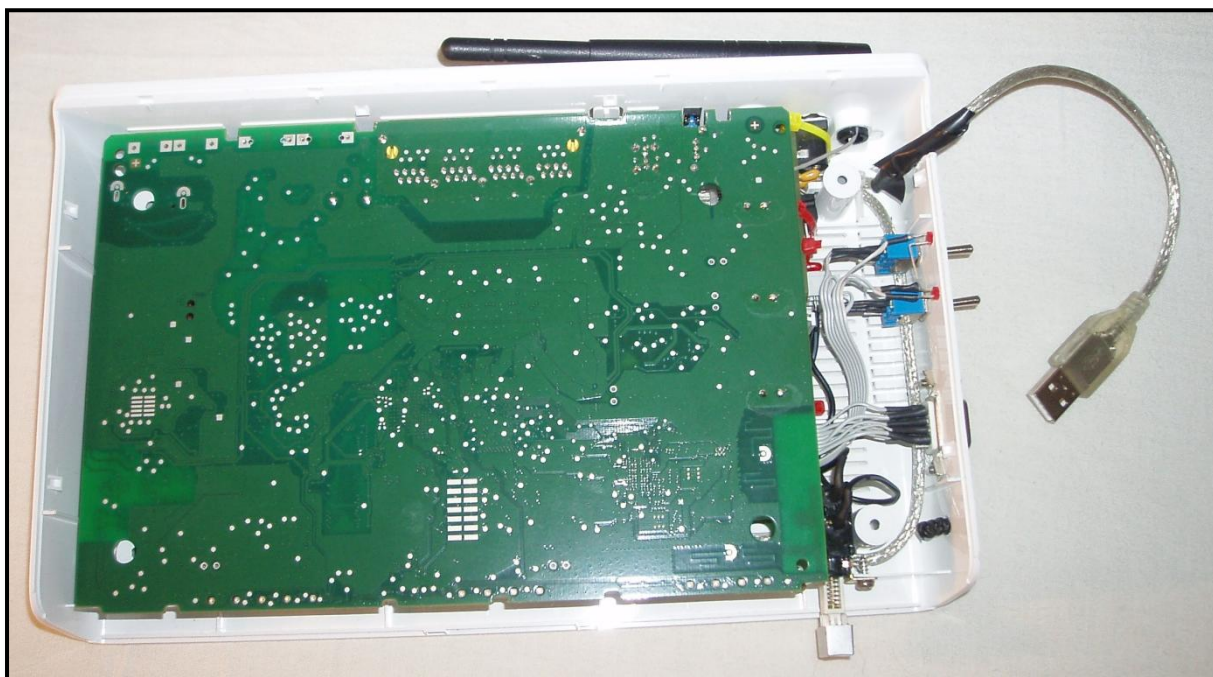
Oznake na slici označuju prethodno opisane elemente i to kako slijedi:

- A – mikrokontroler Arduino
- B – priključna pločica
- C – integrirani krug pl2303
- D – DB9 konektor za priljučak senzora
- E- USB konektor (koji se priključuje u USB port *routera*)
- F – Prekidači, sklopka 1 i sklopka 2
- G – regulator napoma (namijenjen za korekciju napona sa 12V na 9V, za pogon mikrokontrolera)
- H – Sklopka za uključivanje mikrokontrolra



Slika 9-Sučelje između mikrokontrolera i akcelerometra

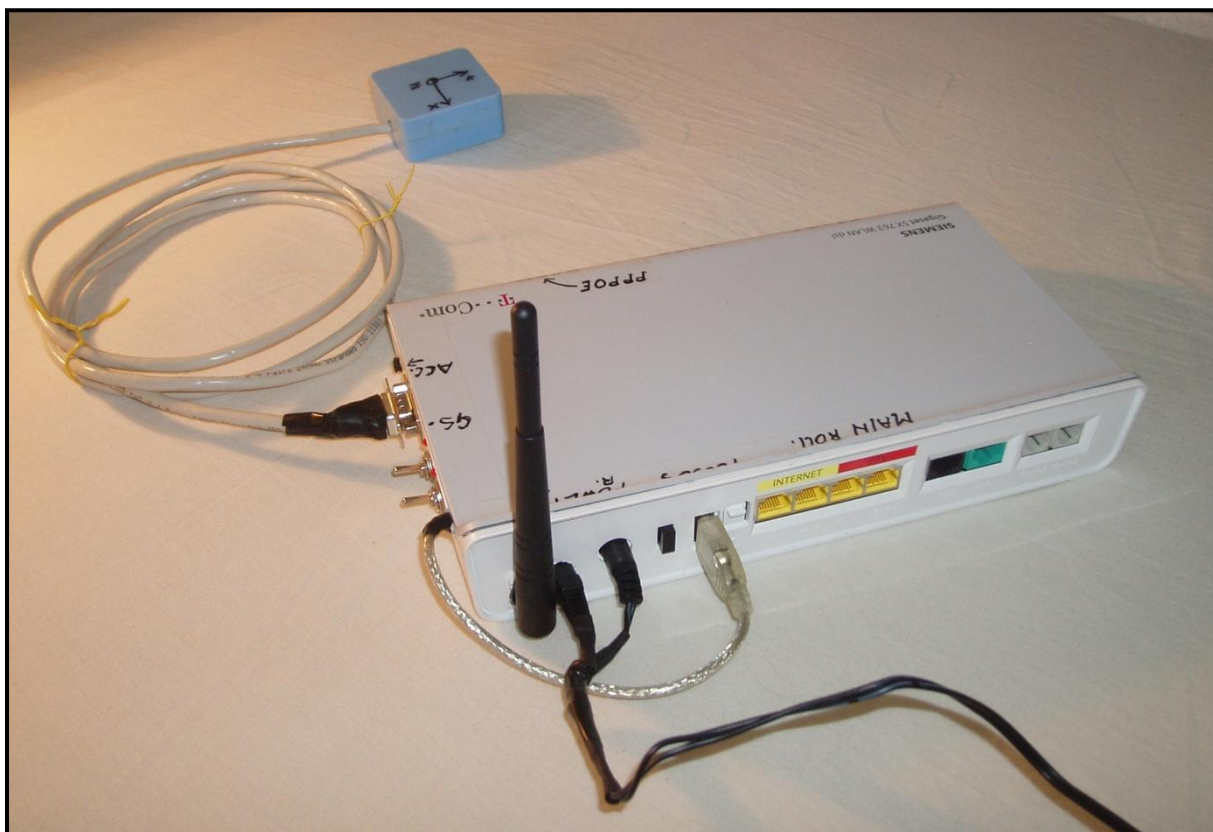
Na slici je prikazan dio sustava koji predstavlja sučelje između akcelerometra i mikronrolera. Fizički jest izveden na pločici koja se fizički priključuje na mikrokontroler, a pogled slike je s donje strane. Veza između mikrokontrolera i routera, jest, kako je ranije opisano[Slika Model fizičkog prototipa], ostvarena *USB-to-serial* pretvornikom, sa pripadajućim priključnim kabelom. Iz mikrokontrolera do kabela veza se ostvaruje putem *RXD* i *TXD* žica, s jedne strane, te s USB priključkom s druge strane. Fizička izvedenica, na prototipu vidljiva je na slici (Slika9).



Slika 10-Kućište sa matičnom pločom routera



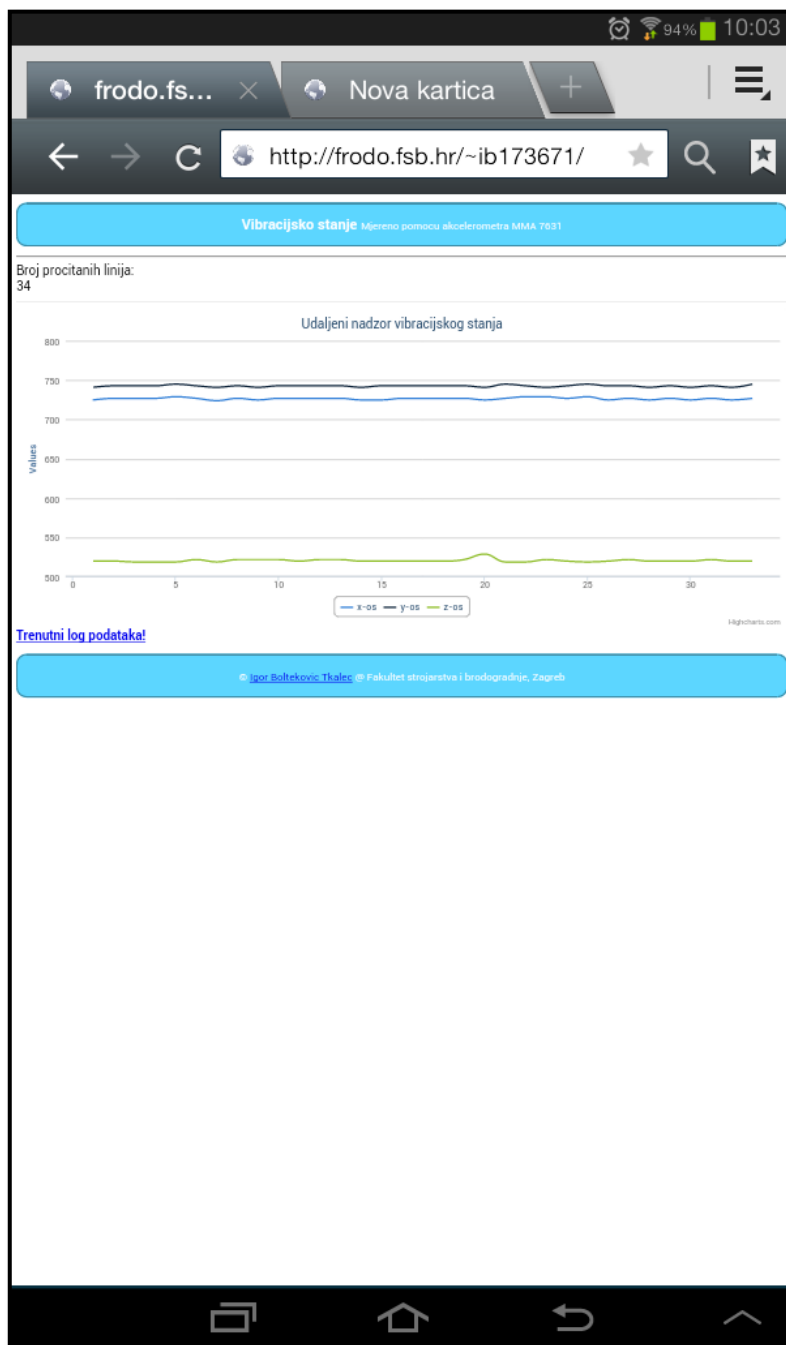
Slika 11-Bočna strana kućišta sa vidljivim konektorom



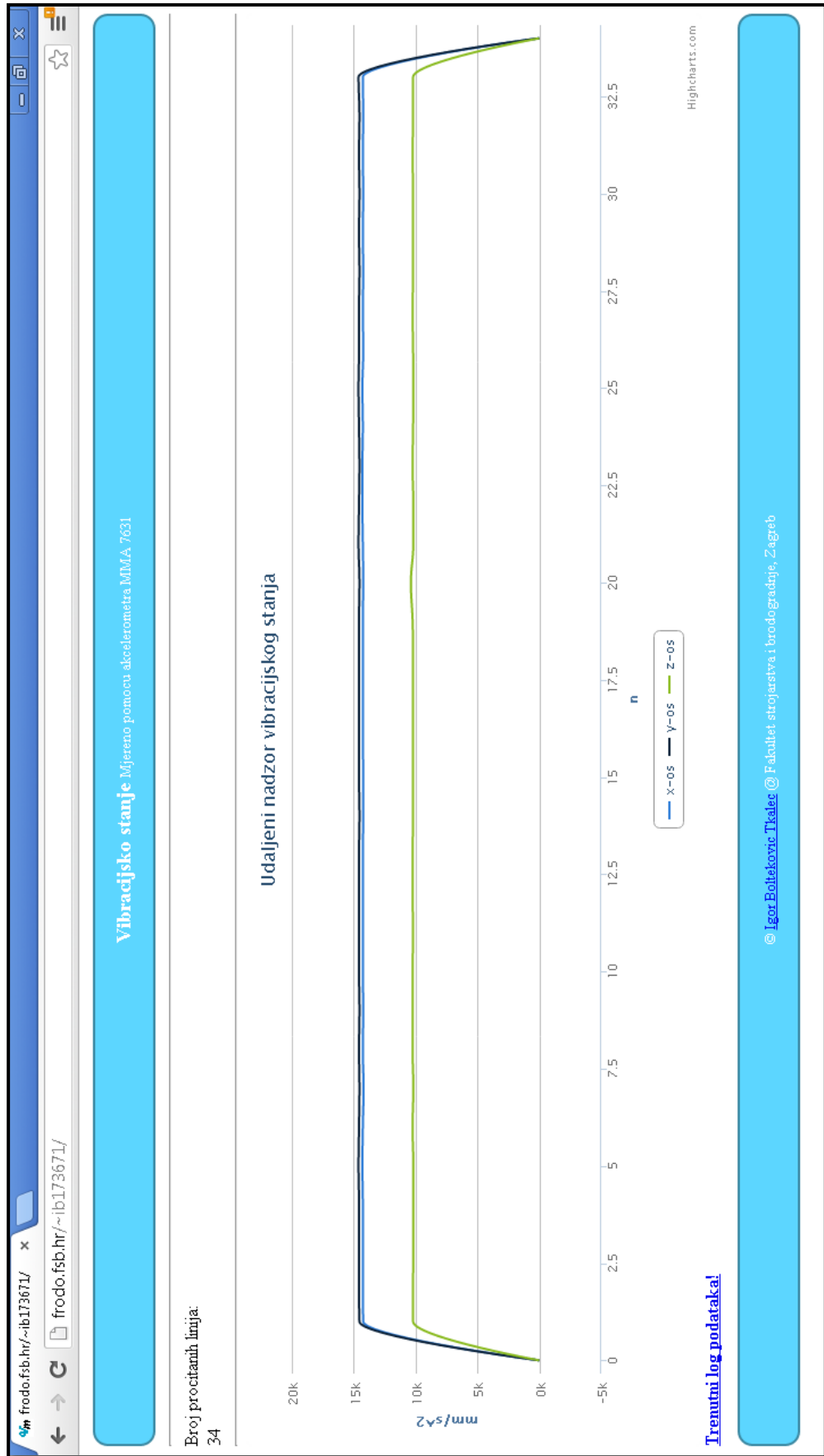
Slika 12-Prototip spreman za korištenje

## 5. Rezultati

Konačni rezultat rada jest grafički i tekstualni pristup podacima mjerenja akcelerometra putem korisničkog preglednika pomoću bilo kojeg uređaja za pristup mrežnim stranicama. Jedini je uvjet da preglednik podržava *JavaScript* programsku podršku, što je danas uvedeno kako standard u sve preglednike, pa čak i one na mobilnim uređajima (*smartphone i tablet računala*). Na naredne dvije zaslonske snimke potvrđuje se gornji navod, to jest prikazuje se demonstracija rada na tablet računalu, putem standardnog *Android* preglednika, te na osobnom računalu, putem preglednika *Mozilla Firefox*.



Slika 13-Zaslonska snimka sustava na android tablet računalu



Slika 14-Zaslonska snimka na osobnom računalu



Da bi se ostvarili rezultati očitavanja uživo potrebno je samo spojiti se SSH konekcijom na server na kojem se nalaze html stranica i php skripta, te iz *shella* pokrenuti skriptu koja otvara mjesto na slobodnom, odabranom portu (naredba *php glavni\_server.php*). Naredni korak jest spojiti se, također SSH konekcijom na router, te izvršiti linijsku naredbu koja će sve primljene podatke proslijediti na prethodno otvoreni port (na adresi glavnog servera) (naredbom *socat /dev/ttyUSB0,nonblock,raw tcp:161.53.116.11:22222*).

Priložene zaslonske snimke zorno pokazuju rezultate mjerenja, s dodatkom da se na poveznici „Trenutni log podaci!“ mogu preuzeti podaci spremni za daljnju analizu u nekom od dostupnih alata. Sam grafički prikaz prikazuje promjenu vibracija u vremenu sa naznakom da se na apscisi nalazi trenutni broj očitavanja dok su na ordinati predstavljene vrijednosti amplitude očitavanja u mjerilu kako je opisano u poglavlju 4.1. to jest svaka jedinica naznačena na grafu označava vrijednost u  $\text{mm/s}^2$  (to jest vrijednost primljena od mikrokontrolera pomnožena sa jest 19,62). Osvježavanje novim podacima izvršava se svakih 300ms, putem ajax protokola integriranog u html stranicu (funkcija js poziva sama sebe svakih 300 ms) sa napomenom da se raspon prikaza prilagođava trenutnim podacima, to jest kad bi primjerice vrijednosti po svim osima bile 500 jedinica, grafički bi se prikaz prilagodio takvim podacima te uvećao raspon oko te vrijednosti. Prelaskom kursora po liniji koja označava pojedinu os očitavanja (x, y ili z) može se očitati točna vrijednost pojedine točke.

Dodatna kontrola nad očitanjima postiže se položajem sklopke 1, to jest njome se upravlja sa osjetljivošću akcelerometra (gornji položaj za 1.5g donji položaj za 6.g)

## 6. Rasprava

Rezultati ovog rada zapravo su samo baza za daljnju integraciju tehnologija, bilo to razvoj novog sustava (sklopovska, *low-level* integracija i mikrokontrolera i senzora i komunikacije), bilo to integracija sa postojećim mjernim uređajima, vibrometrima. Pošto je analiza rezultata mjerenja izuzetno složen zadatak, moguće je i uživa očitavanja povezati sa ekspertnim sustavima koji postoje, a koji bi zajedno sa ovakvim sustavom pružili kompletno automatiziran nadzor nad vibracijskim stanjem, sa ugrađenim znanjem analitičara, a dostupan korisniku putem bilo kojeg modernog uređaja za pregled mrežnih stranica. Takav sustav, baziran upravo na integraciji kakva je prikazana u ovom radu, mogao bi značajno promijeniti poimanje održavanja, kakvo je ono trenutno poznato. Naime, trendovi u održavanju su upravo *e-održavanje*, što podrazumijeva širok spektar tehnologija, a za rezultat ima pravovremeno otkrivanje kvarova, to jest predviđanje kada će se dogoditi trenutak kvara, te se tako održavanje može obaviti upravo kada je potrebno – niti prerano, niti prekasno. Naravno, održavanje prije kvara jest pogodnije, ali opet održavanje se odvija u proizvodnom sustavu, te je želja da se proizvodnja nastavi čim je duže moguće, bez zaustavljanja. Daljnja pogodnost je planiranje održavanja, što ima možda i najznačajniju ulogu u realnom proizvodnom sustavu, jer nema neplaniranog zaustavljanja, što je često podrazumijeva iznimne gubitke i u financijskom smislu, ali i u planiranju proizvodnje.

## **7.Zaključci**

U današnje vrijeme kada su tehnologije u svakom području ljudskog djelovanja razvijene do iznimne razine uspješnosti, vrijeme jest da se čim je više moguće posveti integraciji dostupnih tehnologija kako bi se ostvarile prednosti koje zasebno ne postoje.

Rad je za cilj imao upravo integraciju na više tehnoloških područja, te se može zaključiti kako je osnovni cilj : pristup udaljenim podacima mjerenja vibracijskog stanja uspješno ostvaren. Dakako, rad je zapravo podloga za daljnji razvoj i integraciju, sa napomenom da bi se mogao upotrijebiti ne samo u svrhu preventivnog održavanja, kao u ovom projektu, već da bi se, pošto je sustav rađen modularno, vrlo lako mogli upotrijebiti i drugi senzori, poput jednostavnih: mjerenja temperature, tlaka, GPS lokacije, brzine i slično, ali i bilo kojih složenijih sustava koji za rezultat svojih mjerenja imaju vremenski zavisne podatke. Daljnjim napretkom web tehnologija, bilo bi moguće promijeniti i sučelje, potpomognuto nekim budućim tehnologijama (ili onima u razvoju, koje još nisu opće prihvaćene, ali je izvjesno da će biti).

## **Zahvale**

Zahvaljujem se mentoru prof.dr.sc. Dragutinu Lisjaku, na pruženim smjericama u oblikovanju i ostvarenju ovog rada. Zahvaljujem se i prof.dr.sc. Mladenu Crnekoviću na pruženoj pomoći u smislu fizičkog ostvarenja prototipa. Ovom prilikom se zahvaljujem i općenito svim članovima *open-source* zajednice, koji svoja saznanja daju na uvid zainteresiranima, šireći tako znanja i duh istraživanja.

## Popis literature

[1] MMA7361 Datasheet, Freescale Semiconductor, Denver, Colorado, SAD

[2] Krautov strojarski priručnik, Sajema, Zagreb 2009.g

[3] PL-2303 USB to RS-232 Bridge Controller Product Datasheet , , Taiwan

Web stranice:

<http://www.consultnet.ie/environmental%20vibration.htm>

<http://www.instructables.com/id/Arduino-Microcontroller-Projects/>

<http://grenville.wordpress.com/2012/06/02/an-arduino-uno-talking-to-a-tp-link-wr1043nd/>

<http://stackoverflow.com/questions/6326364/tcp-sending-and-receiving-tcp-ip-data?rq=1>

<http://www.glacialwanderer.com/hobbyrobotics/?p=261>

<http://ajaxpatterns.org/HTTP%5FStreaming>

<http://wiki.openwrt.org/doc/recipes/serialbaudratespeed>

<http://www.highcharts.com/demo/dynamic-update>

<http://docs.highcharts.com/#preprocessing-live-data>

<http://www.binarytides.com/php-socket-programming-tutorial/>

<https://github.com/RobotGrrl/RobotGrrlFFT/blob/master/RobotGrrlFFT.pde>

\*Napomena: web stranicama je pristupljeno višestruko, u razdoblju od prosinca 2012. godine do svibnja 2013. godine

## Sažetak na hrvatskom jeziku

Naziv rada: Udaljeni nadzor vibracijskog stanja stroja

Temelj ovog rada predstavlja integracija i objedinjavanje danas dostupnih tehnologija, u svrhu udaljenog, uživog praćenja vibracijskog stanja mjenenog stroja (ili uređaja) u svrhu preventivnog održavanja. Naime, pravovremeno održavanja proizvodnog sustava otklanja neželjeno, neplansko zaustavljanje rada, te ujedno sve negativne posljedice proizašle iz nenadanih zastoja, a što ponajprije pozitivno utječe na ekonomske aspekte proizvodnje jer uklanja izvanredne troškove i proizvodnom sustavu omogućava dozu sigurnosti u vremenskom, planiranom određivanju troškova vezanih za održavanje proizvodnog sustava. Pojam preventivnog održavanja znači da ćemo funkcionalnost sustava održavati na razini upravo bez zastoja njegovog rada, a mjerenje vibracijskog stanja je upravo jedan od osnovnih načina na koji taj trenutak možemo procijeniti. Sami rad se ne fokusira na obradi prikupljenih podataka, već se fokusira upravo na olakšavanju prikupljanja podataka, na način da se mjerni uređaj fizički približi mjernom mjestu, a rezultate osigura na udaljenoj korisničkoj lokaciji, što znači da analitičar podataka (stručnjak) fizički više ne mora biti prisutan kod mjerenja, što dovodi do mogućnosti *outsourcinga* te tako podiže kvalitetu samog mjerenja, jer je moguće sustav postaviti tako da se eksperti u području više posvete analizi, a ne, do sada vremenski zahtjevnom, prikupljanju podataka.

Rezultat rada jest funkcionalni prototip udaljenog sustava nadzora vibracijskog stanja, a sastoji se od senzora vibracija (akcelerometar), koji mjeri promjenu ubrzanja po svim osima, mikrokontrolera koji prikuplja podatke senzora, šalje ih mrežnom uređaju (*routeru*), koji pak tako pripremljene podatke putem svjetske mreže računala šalje na udaljeni server, na kojem se pak podaci pripremaju za ispis na korisnički zahtjev putem web preglednika. Na samom korisničkom dijelu sustava podaci su dostupni u tekstualnom obliku pogodnom za daljnju analizu, te isto tako i uživo i putem grafičkog prikaza. Rad integrira tehnologije dostupne na polju prikupljanja, obrade i slanja podataka, dok je neovisan o korisničkoj platformi kojom se pristupa rezultatima, te upravo zbog prirode samog sustava, korisnik je u mogućnosti koristiti sve danas dostupne uređaje za pristup mreži (kućno računalo, prijenosnik, tablet računalo ili *smartphone* uređaje). U drugu ruku, ovakav sustav predstavlja inovativan pristup, te se nadogradnjom mogu postići i druge svrhe mjerenja, jer je praktički nebitno kakav se senzor s jedne strane koristi, te koju namjenu ovako postavljeni sustav u konačnici ima.

Ključne riječi, pojmovi: udaljeni nadzor, uživo prikupljanje podataka, vibracijsko stanje, mrežne tehnologije

## **Summary**

Title: Remote, real-time monitoring of vibrational state of the machine

The basis of this work is the integration and unification of currently available technologies for remote, real time monitoring of vibrational states of measured machine (or device) for preventative maintenance. Specifically, preventative maintenance of the production system removes unwanted, unplanned downtime, and also all the negative consequences arising from the sudden halt, which primarily has a positive effect on the economic aspects of the production because it eliminates the extra costs and production system provides peace of mind in the planning, due to determining costs related to maintenance of the production system. The concept of preventive maintenance means that we will maintain the functionality of the system, precisely, without delay of operation, and measure the vibrational state is just one of the main ways in which this moment can be estimated. This research paper does not focus on the processing of collected data, but focuses just on facilitating data collection, so that the measuring device is physically closer to the measurement point, and the results provided for remote user location, which means that the data analyst (expert) physically no longer has to be present in the measurement, which leads to the possibility of outsourcing and thus improves the quality of the measurements. It is possible to set up the system so that experts in the field focuses to the analysis, and not, like so far, on time-consuming data collection.

The result of research paper is a functional prototype of a remote monitoring system vibrational state, and consists of a vibration sensor (accelerometer), which measures the change in acceleration in all axes, a microcontroller that collects sensor data, sends them to a network device (router), which then sends that the data via the world network computer web sends to a remote server, which prepares data for printing on user demand through a web browser. On user side data is available in text format suitable for further analysis, and also live (real-time) through a graphical display(line graph). The work integrates technology available in the field of collecting, processing and transmitting data, and is independent of the user platform for approach and because of the nature of the system, the user is able to use all available devices for network access (home computer, notebook, tablet computer or smartphone). On the other hand, this system represents an innovative approach, and with upgrade, other purposes of measurement can be achieved and it is almost irrelevant what type of sensor is used on one side, and what purpose such a system installed, on the other side,has

Key words: remote monitoring, live data collection, vibrational state, network technology

## **Životopis**

Rođen sam u Koprivnici, 1989.godine., te od onda živim u Đurđevcu, gdje sam pohađao osnovnu školu, te kasnije gimnaziju (2004.-2008.). Po završetku gimnazijskog obrazovanja upisujem Fakultet strojarstva i brodogradnje u Zagrebu. Titulu prvostupnika strojarstva stječem 2012. godine, te nastavljam školovanje na istom smjeru (Industrijsko inženjerstvo i menadžment), sve do danas. Dobitnik sam nagrade za najbolji rad u biologiji na državnom natjecanju 2008.godine (tema:eksperimentalna proizvodnja biodiesela). U slobodno vrijeme rekreativno se bavim biciklizmom, elektronikom, programiranjem.



## Prilog 1.

U prilogu se nalazi programski kod(sa karakteristikama: naziv programa, naziv jezika, naziv komponente na kojoj se koristi) svake od komponente sustava prema sljedećoj listi:

- a) Programski kod mikrokontrolera (*sketch*)
- b) Popis programskih paketa mrežne komponente sustava (router) nužnih za ostvarenje komunikacije opisane u Shemi modela sustava, te linijski programski kod za pokretanje komunikacije
- c) Programska skripta na strani servera (poslužitelja) zadužena za ostvarivanje komunikacije i pripreme podataka za korištenje iz preglednika
- d) Web stranica za korištenje na pregledniku, zadužena za ispis (tekstualni i grafički)

a) Programski kod mikrokontrolera

Naziv mikrokontrolera: Arduino UNO R3

Naziv programa (*sketcha*): MjerenjeVibracija\_web2.ino

Naziv jezika: *Arduino IDE (integrated development environment)*, koji je inačica *Wiring* platforme, čiju sintaksu i biblioteku i koristi, a izuzetno je sličan C++ programskom jeziku, sa pojednostavljenjima i modifikacijama

Programski kod:

```
//Sketch za slanje ocitanja s akcelerometra, na serijski port,
jednostavnije
//Igor Boltekovic Tkalec, @FSB, Zagreb
//postavljanje konstanti za prikljuceni "shield"
int led1 = 13;
int led2 = 12;
int sklopka1 = 11;
int sklopka2 = 10;
int Gselect = 9; //Pin na kojem je voltae divider da daje 3.3V za kontrolu
sensitivityja na acc
int sklopka1Stanje = 0; //varijabla za citanje stanja sklopke1
int sklopka2Stanje = 0;
const int xPin = A3; // x-axis of the accelerometer
const int yPin = A2; // y-axis
const int zPin = A1; // z-axis (only on 3-axis models)
int xAcc=0; //Postavlja varijable za ocitanja akeleracije
int yAcc=0;
int zAcc=0;
int outputXAcc=0;
int outputYAcc=0;
int outputZAcc=0;

int loopCount = 12; //Postavlja ciklus, koliko ce se ocitanja u jednom
loopu uzeti za srednju vrijednost
void setup()
{
  pinMode(led1, OUTPUT); //Postavljanje ledica kao output pin
  pinMode(led2, OUTPUT);
  pinMode(sklopka1, INPUT_PULLUP); //Postavljanje sklopke1 kao input pin
  pinMode(sklopka2, INPUT_PULLUP); //Postavljanje sklopke2 kao input pin
  pinMode(Gselect, OUTPUT); //Postavlja pin 9 kao output, za kontrolu
sensitivityja acc
  digitalWrite(Gselect, LOW);
  //Postavlja serijski port
  Serial.begin(9600);
}
void loop()
{
  int xAcc=0;yAcc=0;zAcc=0; //Definiranje varijabli za spremanje ocitanja,
u loopu da se pobrisu Pri svakom novom mjernju
  //UMETNUTO-ZA-UPRAVLJANJE-SKLOPKAMA-----
  sklopka1Stanje = digitalRead(sklopka1);
  // Provjera da li je sklopka1 HIGH ili LOW , ako je HIGH ukljuci LED1

  if (sklopka1Stanje == HIGH) {
```

```
// turn LED on:
digitalWrite(led1, LOW);
//Serial.println("UKLJUCENA SKLOPKA 1");
digitalWrite(Gselect, LOW);
}

else {
// turn LED off:
digitalWrite(led1, HIGH);
digitalWrite(Gselect, HIGH);
}

sklopka2Stanje = digitalRead(sklopka2);
// Provjera da li je sklopka2 HIGH ili LOW , ako je HIGH ukljuci LED2
if (sklopka2Stanje == HIGH) {
// turn LED on:
digitalWrite(led2, LOW);
//Serial.println("UKLJUCENA SKLOPKA 2");
}
else {
// turn LED off:
digitalWrite(led2, HIGH);
}
//Jednostavan filter (uzima srednju vrijednost od 12 mjernja!)
for(int i = 0; i< loopCount; ++i)
{
// Treba 100 us (0.0001 s) da se procita analog input
xAcc += analogRead(xPin);
yAcc += analogRead(yPin);
zAcc += analogRead(zPin);
}

xAcc /= loopCount;
yAcc /= loopCount;
zAcc /= loopCount;

//Mapira ulaz iz 0 do 1023 na 0 do 255
outputXAcc=map(xAcc,0,1023,0,255);
outputYAcc=map(yAcc,0,1023,0,255);
outputZAcc=map(zAcc,0,1023,0,255);

// print the sensor values:
Serial.print(outputXAcc);
// print a tab between values:
Serial.print("\t");
Serial.print(outputYAcc);
// print a tab between values:
Serial.print("\t");
Serial.print(outputZAcc);
Serial.println();
// delay before next reading:
delay(10);
}
```

b) Popis programskih paketa mrežne komponente sustava (*router*)

Naziv *routera*: Siemens Gigaser SX 763 WLAN dsl

Inačica operativnog sustava to jest *firmware-a*:

OpenWRT, (*openwrt-lantiq-danube-GIGASX76X-squashfs.image*),

verzija : *Barrier breaker (Bleeding Edge, r33854)*,

verzija linux kernela: 3.3.8.

Popis programskih paketa nužnih za funkcioniranje prema modelu:

Kmod-usb-acm

Kmod-usb-core

Kmod-usb-dwc-otg

Kmod-usb-ohci

Kmod-usb-serial

Kmod-usb-pl2303

Usbutils

Socat

Ser2net

Linijaska naredba za omogućavanje komunikacije između pl2303 i servera:

```
socat /dev/ttyUSB0 tcp:161.53.116.11:22222 (šelje Cr i lf)
socat /dev/ttyUSB0,nonblock,raw tcp:161.53.116.11:22222
```

Jednostavna provjera da li se podaci primaju putem serijske veze na strani routera

```
dmesg //provjera na kojem je portu prisutan USB uređaj
cat echo /dev/ttyUSB0 //ako je uređaj bio prisutan na ttyUSB0 kao u konkretnom uređaju
CTRL+C //terminacija čitanja
```

Programska skripta na strani servera (poslužitelja) zadužena za ostvarivanje komunikacije i pripreme podataka za korištenje iz preglednika

Naziv servera: *frodo.fsb.hr*

Naziv programskog jezika: PHP (*Hypertext Preprocessor*), skriptni jezik serverske strane sustava

Naziv skripte: *glavni\_server.php*

Linijaska naredba za pokretanje skripte (mora se pokrenuti iz *shell*a servera):

```
cd /putanja do skripte/  
php glavni_server.php
```

Programski kod skripte:

```
<?php  
/*Skripta za "slušanje" porta na zadanoj adresi, a primljene podatke šalje  
prema html fajlu ()  
nalazi se u u  
cd htdocs/vibracije  
  
Schema: [router]----- (tcp/ip)>ova_skripta.php-----  
(websocket)>html.fajl  
Pseudokod:  
    stvori tcp socket na addr i port (za komunikaciju sa routerom)  
    poveži tcp socket sa adresom koja se spojila  
    slušaj na povezanom socketu  
  
    stvori websocket (za komunikaciju sa browserom)  
    poveži websocket  
    pošalji podatke prikupljene sa tcp socketom na browser putem  
websocketa  
  
    ponavljaj gornje korake  
  
*/  
  
error_reporting(~E_NOTICE);  
set_time_limit (0); //Osigurava da server ne ode u time-out  
  
//-----TCP---SOCKET-----  
-----  
//Definiranje varijabli  
$adresa = "127.0.0.1"; //(localhost)  
//Dok skripta bude na webu, uzmi adresu servera  
//$adresa = $_SERVER['SERVER_ADDR'];  
$port = 5000; //proizvoljno odabrani, nezauzeti port  
$podaci =array(); //definiranje varijable za pohranu primljenih podataka.  
$buffer_size = 1024; //Velicina buffera (koliko podataka max mogu  
primiti), vrijedost je u bajtovima ; za pocetnu vr odabrano 1024, paziti  
kod testiranja!  
$server_poruka = "";
```

```
$temp_file="trenutni_podaci.txt"; //Svojevrnsni buffer file za trenutne, raw
podatke
$log_file="log.txt"; //Datoteka u koju se spremaju svi primljeni podaci,
zajedno za timestampom, kada je podatak primljen
$i=0;

//Definiranje varijabli i funkcija potrebnih za proceduru "socket"-a
//Nalaze se u if then, zbog mogucih gresaka

//Stvori socket, ako ne uspijes, javi gresku
//AF_UNIX, SOCK_DGRAM, 0); ili AF_INET, SOCK_STREAM, 0 -radi ili
AF_INET, SOCK_RAW, getprotobyname('udp')
if(!($sock = socket_create(AF_INET, SOCK_STREAM, 0))//Af_inet:IPv4 tip,
SOCK_STREAM:full duplex, TCP protokol
{
    $errorcode = socket_last_error();
    $errormsg = socket_strerror($errorcode);

    die("Couldn't create socket: [$errorcode] $errormsg \n");
}
echo "Socket stvoren! \n";

//Povezi otvoreni socket sa zadanom adresom, ako ne uspijes, javi gresku
if( !socket_bind($sock, $adresa, 5000) )
{
    $errorcode = socket_last_error();
    $errormsg = socket_strerror($errorcode);

    die("Could not bind socket : [$errorcode] $errormsg \n");
}
echo "Socket povezan! \n";

//Slusaj predodno otvoreni socket, ako ne uspijes, javi gresku
if(!socket_listen ($sock , 10))
{
    $errorcode = socket_last_error();
    $errormsg = socket_strerror($errorcode);

    die("Could not listen on socket : [$errorcode] $errormsg \n");
}
echo "Socket slusa! \n";

//Prihvati pridodlog klijenta na socketu koji si slusao predhodno
$trenutni_klijent = socket_accept($sock);

//Odmah napravi konekciju na pridodlog klijenta na otvorenom socketu!
koristiti ili socket_accept(gore) ili socet_connect(dolje)
//$trenutni_klijent=socket_connect($sock, $adresa, $port);

while(true){

//Procitaj sto prihvaceni klijent ima od podataka
//Bilo je $podaci_klijenta = socket_read($trenutni_klijent,$buffer_size);
//socket_read($trenutni_klijent,11,PHP_NORMAL_READ); Radi i s ljepsi ispis
ali jako sporo
//$podaci_klijenta = socket_read($trenutni_klijent,11,PHP_NORMAL_READ);
$linija = socket_read($trenutni_klijent,11,PHP_NORMAL_READ);
$podaci[$i] = $linija."\n";
```

```
//prekida ako je doslo do pograske
if(!socket_read($trenutni_klijent,11,PHP_NORMAL_READ))
{
    $errorcode = socket_last_error();
    $errormsg = socket_strerror($errorcode);

    die("Doslo je do greske, ocito je klijent napustio socket \n");
}

if($i>100){
    file_put_contents($temp_file,$podaci);
    file_put_contents($log_file,$podaci,FILE_APPEND | LOCK_EX);

    unset($podaci);
    $i=0;
    usleep(30);
}
//echo trim($podaci[$i])."\r";
$i++;
}
//-----TCP---SOCKET---KRAJ-----
-----

socket_close($sock);
```

d) Web stranica za korištenje na pregledniku, zadužena za ispis (tekstualni i grafički)

Naziv programskog jezika: integrirano HTML (*HyperText Markup Language*), JavaScript, JQuery

Naziv stranice: *index.html*

Programski kod:

```
<!DOCTYPE html>
<html>
<head>
<!--<script
src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script language="javascript" src="js/init.js"></script>
<script language="javascript" src="js/dsp.js"></script>
<script src="js/highcharts.js"></script>
<script src="js/exporting.js"></script> -->
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.2/jquery.min.js"></scr
ipt>
<script src="http://code.highcharts.com/highcharts.js"></script>
<meta name="description" content="Web aplikacija za pracenje vibracijskog
stanja mjenenog pomocu akcelerometra, arduina, routera">
<link rel="stylesheet" href="css/superfish.css" media="screen">

</head>
<body onload="ucitajPodatke()">
<h1 id="banner">Vibracijsko stanje <small>Mjereno pomocu akcelerometra MMA
7631</small></h1>
<!--Izdvojena samo jedna linija:
<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>
<hr>
Sve procitane linije:
<div id="div2"><h2>text</h2></div>
<hr>
Izdvojena jedna tocka iz linije:
<div id="div3"><h2>text</h2></div>
<hr>
Izdvojena jedna tocka iz linije:
<div id="div4"><h2>text</h2></div>
x-os:
<div id="div4"><h2>text</h2></div>-->
<hr>

Broj procitanih linija:
<div id="div5"><h2>text</h2></div>
<hr>
<div id="container" style="height: 400px"></div>

<!--<script>
var chart = new Highcharts.Chart({
  chart: {
    renderTo: 'container'
  },
  series: [{
    data: [29.9, 71.5, 106.4, 129.2, 144.0, 176.0, 135.6, 148.5, 216.4,
194.1, 95.6, 54.4]
  }]
})
```



```

}); -->

</script>
<script>
var chart = new Highcharts.Chart({
  chart: {
    type: 'spline',
    renderTo: 'container'
  },
  title: {
    text: 'Udaljeni nadzor vibracijskog stanja'
  },
  plotOptions: {
    series: {
      marker: {
        enabled: false
      }
    }
  },
  yAxis: {
    title: {
      text: 'mm/s^2'
    }
  },
  xAxis: {
    title: {
      text: 'n'
    }
  },
  series: [{
    name: 'x-os',
    data: [129.9, 171.5, 106.4, 129.2, 144.0, 176.0, 135.6, 148.5,
216.4, 194.1, 95.6, 540.4]
  },{
    name: 'y-os',
    data: [129.9, 171.5, 106.4, 129.2, 144.0, 176.0, 135.6, 148.5,
216.4, 194.1, 95.6, 540.4]
  },{
    name: 'z-os',
    data: [129.9, 171.5, 106.4, 129.2, 144.0, 176.0, 135.6, 148.5,
216.4, 194.1, 95.6, 540.4]
  }
  ]
});

// the button action
//$('#button').click(function() {
//  chart.series[0].setData([129.2, 144.0, 176.0, 135.6, 148.5, 216.4,
194.1, 95.6, 54.4, 29.9, 71.5, 106.4] );
//});

//-----
//-----
var data=[]; // array za citanje cijele datoteke trenutni_podaci.txt
var data_raw=""; //var za citanje podataka kako su primljeni
var lines=[]; // array za pohranu pojedine linije
var red=[];
var tocka=[]; //array za izdvajanje pojedine tocke
var redStr=new String();
var k=0.01962; //koeficijent pretvorbe u m/s^2 prema radu.

```

```

//var k=19.62; // koeficijent pretvorbe u mm/s^2 prema radu
//var x_os=[];
//var y_os=[];
//var z_os=[];

function ucitajPodatke() {
//chart.series[0].setData([29.9, 71.5, 106.4, 129.2, 144.0, 176.0, 135.6,
148.5, 216.4, 194.1, 95.6, 54.4] );
var x_os=[];
var y_os=[];
var z_os=[];
var x_os_str="";
//$(document).ready(function() {
//$("#button").click(function() {
$.ajax({url:"trenutni_podaci.txt",success:function(result) {
//data.clear;tocka.clear;
var data=result.split(/\r/); //U array data su sada spremljene
pojedine linije

var red=new Array(data.length);
for (var i=0;i<data.length;i++)
{
red[i]=new Array(3);
red[i]=data[i].split(";");
redStr=red[i];//.toString();
for (var j=0;j<3;j++){
red[i][j]=redStr[j];//.split(";");
}
}

for (var k=0;k<data.length;k++)
{
x_os[k]=getNum(parseInt(red[k][0]))*19.62;
y_os[k]=getNum(parseInt(red[k][1]))*19.62;
z_os[k]=getNum(parseInt(red[k][2]))*19.62;
}

function getNum(val)
{
if (isNaN(val))
return null;
else
return val;
}

var x_os_str="["+x_os.join(", ")+"]";
//red=data[0].split(";");//Petlja za izdvajanje svih linija u array
//redStr=red[0].toString();
//red2=data[1].split(";");
//$("#div1").html(lines[0]);
//$("#div1").html(data[3]);
//$("#div2").html(data+" linija");
//$("#div3").html("x: "+red[0][0]+",y: "+red[0][1]+", z:
"+red[0][2]);
//$("#div4").html(red[25][2]);
$("#div5").html(data.length-1);
//$("#div4").html(eval(x_os));
//chart.series[0].addPoint(x_os, true,true);//, false);
//chart.series[1].addPoint(y_os, true,true);//, false);
//chart.series[2].addPoint(z_os, true,true);//, false);

```

```
chart.series[0].setData(x_os,true);
chart.series[1].setData(y_os,true);
chart.series[2].setData(z_os,true);
setTimeout(ucitajPodatke, 300); //Fja poziva samu sebe nakon 100 ms

//chart.series[0].setData([129.2, 144.0, 176.0, 135.6, 148.5,
216.4, 194.1, 95.6, 54.4, 29.9, 71.5, 106.4] );
    });
//});
//});
}

</script>
<div><a href="log.txt"><b>Trenutni log podataka!</b></a> </div>
<h1 id="footer"> <small>&copy; <a
href="mailto:ib173671@stud.fsb.hr?Subject=Mjernje vibracija, site">Igor
Boltekovic Tkalec</a> @ Fakultet strojarstva i brodogradnje, Zagreb</small>
</body>
</html>
```

## **Prilog 2.**

Prilog 2 jest DVD-R medij, na kojem se nalaze:

- ovaj rad u pdf formatu
- zaslonska snimka (video) rada sustava
- programski kod, te programske datoteke spomenute u radu
- fotografije sustava