Komunikacija između mikroupravljača primjenom Ethernet protokola

Tomas, Nikola

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

Komunikacija između mikroupravljača primjenom Ethernet protokola

Završni rad

Nikola Tomas

Osijek, 2022.

SADRŽAJ

1. UVOD

Danas mikroupravljači uvelike brojčano premašuju svjetsku populaciju te svoje domove vjerojatno dijelimo s njih pedesetak. Ima ih u svakoj bitnoj industriji kao što su tehnološka, prehrambena, proizvodna, industrija transporta, komunikacije, medicina i ostalo. Ustvari je jako malo stvari koje se danas ne oslanjaju na mikroupravljač.

Mikroupravljač koji ćemo koristiti tijekom ovog završnog rada bit će iz obitelji Arduino mikroupravljača. Arduino je *open-source* računalna i softverska platforma pomoću kojega se stvaraju uređaji koji omogućuju spajanje računala sa fizičkim svijetom. Arduino je jako brzo od svog osnutka postao neizostavan alat za rad s elektronikom i programiranjem. Zbog niske cijene i jednostavnosti implementacije, možemo ga pronaći u mnogim obrazovnim ustanovama. Budući da je cilj ovog rada komunikacija između mikroupravljača potreban nam je Ethernet protokol. Ethernet je vrsta mrežne tehnologije koja se koristi za izgradnju LAN mreža. Izabrali smo Ethernet komunikaciju zbog njene jednostavnosti, velike raširenosti te zbog brzine komunikacije. Arduino također, osim mikroupravljačkih pločica nudi i razne dodatke za njih. Tako nam za ovaj rad odgovara *Ethernet Shield* pločica koja ujedno sadrži sve potrebne priključke za ostvarivanje Ethernet komunikacije. Za korisnike koji se služe sa Arduino mikroupravljačkim pločicama Arduino je napravio svoj vlastiti programski paket za programiranje, kompajliranje (eng. *Compiling*) i prebacivanje programskog koda tj. programa na mikroupravljač. Budući da se komunikacija odvija preko Ethernet protokola koristit ćemo dva mikroupravljača te će komunikacija biti izvedena preko arduino serijskog monitora (*eng. Serial Monitor*).

U poglavlju dva smo opisali sve komponente. Opisali smo ulogu i način rada Arduino Nano mikroupravljača i Arduino Ethernet Shield koji je dodatna komponenta potrebna za Ethernet komunikaciju. Objasnili smo softwerski dio Arduino IDE koji dolazi uz sve Arduino pločice te smo pojasnili njegov način rada kao i što je Ethernet komunikacija i kako se ostvaruje te koje komponente zahtjeva. U trećem poglavlju smo pokazali te objasnili fizičku i programsku izvedbu komunikacije dok smo u zaključku iznijeli koja smo sve znanja stekli izvedbom ove komunikacije.

1.1. Zadatak završnog rada

U ovom završnom radu je potrebno primijeniti ethernet modul te uspostaviti komunikaciju između računala i mikroupravljača.

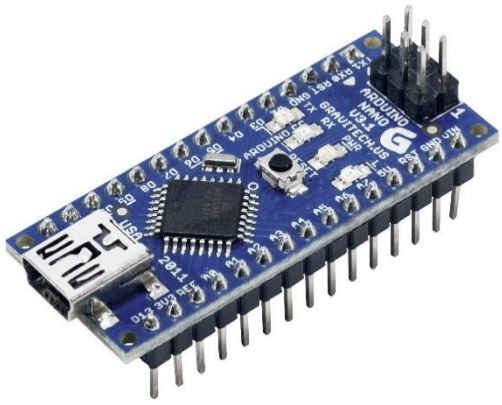
2. PRIMIJENJENE TEHNOLOGIJE I KOMPONENTE

2.1. Arduino Nano

Arduino mikroupravljači su 2005. godine napravljeni kao platforma namijenjena u edukativne svrhe. Zašto je danas jedan od najpopularnijih na svijetu govori nam ovih pet razloga uspješnosti:

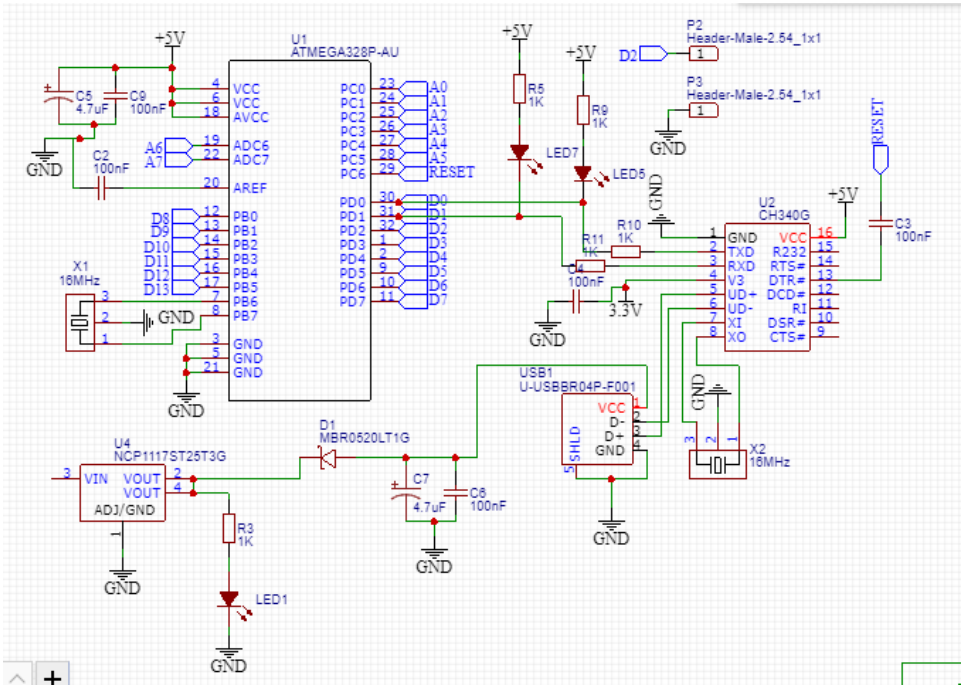
- Moguće ga je nabaviti za manje od 30€ , znači da je financijski dostupan
- Spaja se USB-om, koristi standardne protokole
- Jednostavnost
- Arduino razvojno okruženje dolazi skupa s njim
- Velik broj primjera, kodova, projekata
- Fleksibilnost

U ovom završnom radu ćemo koristiti dvije Arduino Nano izvedbe ovih mikroupravljača. Jedina razlika između dvije Nano ploče je upravljanje, jedan se služi čipom ATmega328P a drugi ATmega168. [4]



Slika 2.1. Arduino Nano

Arduino Nano je mikroupravljačka ploča kojoj je najbitniji dio mikročip koji njime upravlja, a to je ATmega328P odnosno ATmega168. Arduino Nano sklop sadrži 8 ulaznih analognih pinova te 14 ulazno/izlazna digitalna pina, USB i tipku za reset, ICSP zaglavlje tj. konektor koji predstavlja jedan od načina spajanja za vrijeme programiranja Arduina. Budući da ima u sebi USB priključak jednostavno se spaja na računalo pomoću USB kabla. Frekvencija pri kojoj radi pločica je 16 MHz. Arduino pločica nema ugrađene potrebne izvode za RJ-45 (UTP kabel s 8-polnim modularnim utikačem) priključak, te je zbog toga za povezivanje potrebno smisliti dodatno rješenje. [1]



Slika 2.2 Arduino Nano shematski dijagram

Mikroupravljač koristi 32 KB flash memorije, odnosno 2 KB koristi za pokretanje radnog sustava. Sadrži 2 KB SRAM-a i 1 KB EEPROM memorije. Svaki od digitalnih pinova može se koristiti kao ulaz (*eng. input*) i izlaz (*eng. output*). Naredbe pomoću kojih se postiže ta konfiguracija:

- digitalRead() – Čita HIGH i LOW stanje ulaznog pin-a
- digitalWrite() – Postavlja HIGH i LOW stanje pina
- pinMode() – postavlja način rada pina (INPUT, OUTPUT ili INPUT_PULLUP)

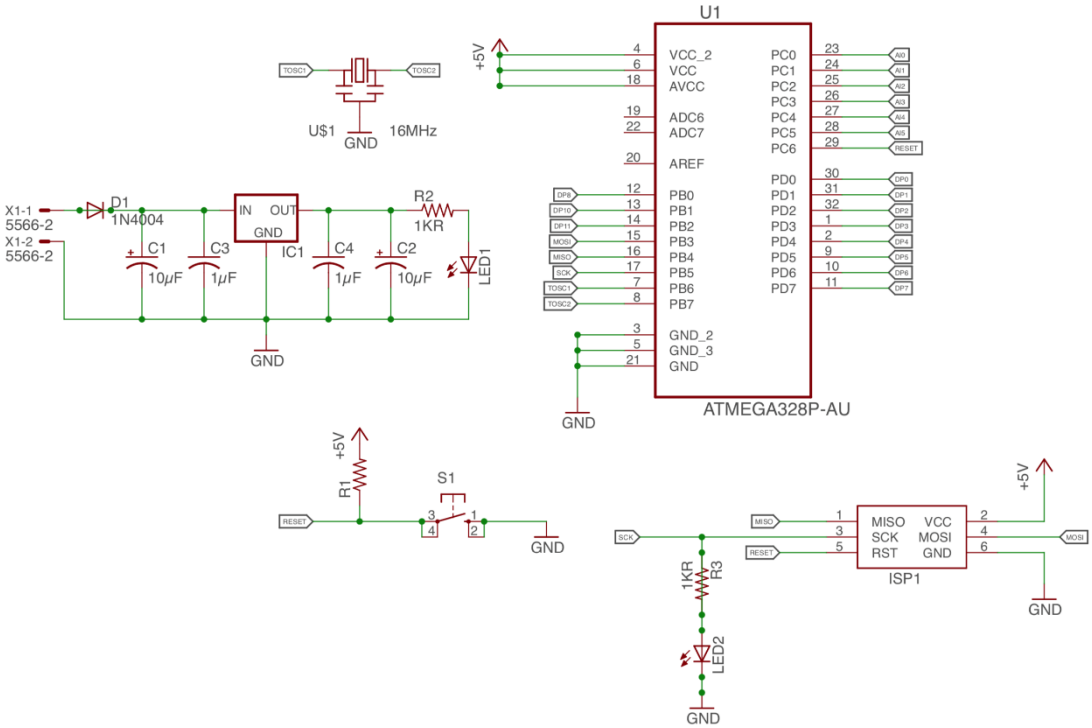
Arduino Nano sadrži niz objekata koji služe za komunikaciju s računalom, drugim mikroupravljačima ili drugom Arduino pločicom. ATmega328P čip nam omogućuje serijsku komunikaciju. Za slanje i primanje jednostavnih tekstualnih podataka Arduino softver koristi serijski monitor. Svjećica TX LED i RX prilikom prijenosa podataka USB-om ili putem FTDI čipa će treptati ali ne i na pinovima 0 i 1 za serijsku komunikaciju. Koriste se pri naponu od 5V. [2] [3]

Mikrokontroler	ATmega168/ATmega328P
Operativni napon	5 V
Preporučeni ulazni	7 V – 12 V
Granični ulazni napon	6 V – 20 V
Digitalni U/I pinovi	14
Analogni ulazni pinovi	8
DC struja U/I pinovi	20mA
DC struja na 3.3V pinu	50mA
Flash memorija	32 KB
SRAM	2 KB
EEPROM	1KB
Takt procesora	16 MHz
Veličina	1.85cm x 4.3cm

Tab. 2.1. – Specifikacije Arduina Nano

Najbitniji dio Arduina, kao što smo već spomenuli, je 8-bitni AVR mikroupravljač visokih performansi i niske potrošnje energije. Može se pohvaliti dobrim performansama zahvaljujući na svojoj naprednoj RISC arhitekturi koja se temelji na procesorima sa manjim opsegom naredaba povećanjem broja registara dostupnim CPU, stavljanjem priručnih memorija na CPU te omogućavanje izvršenja više naredaba unutar jednog otkucaja unutarnjeg sata procesora. [8]

Slika 2.3. Izgled mikročipa ATmega328P



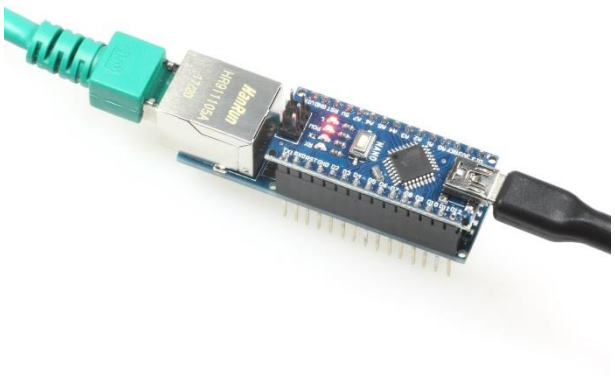
Slika 2.4. Shematski dijagram ATmega328P

2.2. Arduino Ethernet Shield

Arduino Ethernet Shield je dodatna Arduino ploča koja se spaja na vrh Arduino mikroupravljača te mu proširuje mogućnosti sa sklopovima za povezivanje na mrežu korištenjem običnog Ethernet kabla. Drugim riječima omogućava jednostavno povezivanje na internet. Za ovaj rad smo koristili ENC28J60 Ethernet Shield za Nano pločicu. ENC28J60 je samostalni Ethernet kontroler sa SPI (*eng. Serial Peripheral Interface*) sučeljem, ugrađenim MAC (*eng. Media Acess Control*) koji ima zadaću upravljati pristupom mediju i PHY (*eng. Ethernet physical layer*) ili fizički sloj koji je zadužen za prijenos podataka koji se odvija, bit po bit , preko fizičkog medija. Ima ugrađeno i 8 KB RAM-a te zadovoljava sve specifikacije IEEE 802.3 standarda. IEEE 802.3 definira fizički i podatkovni sloj mreža poznatiji kao Ethernet. [5] [7]

Značajke Nano Ethernet Shielda :

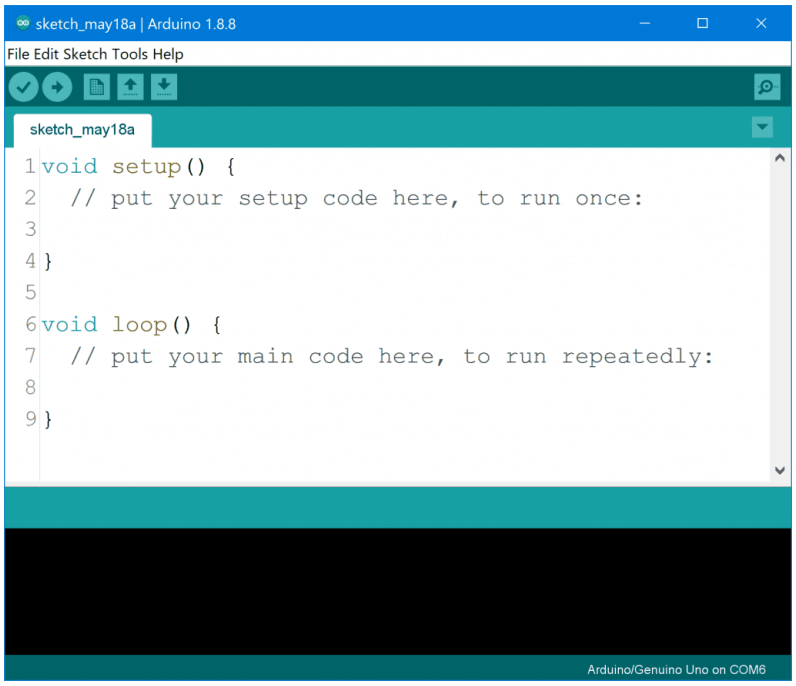
- Integrirani MAC i 10Base-T PHY
- Podržava full i half-duplex modove
- Podržava unicast, multicast i brodcast pakete
- Konfigurabilna veličina međuspremnika prijenosa/prijema
- SPI sučelje s brzinama takta do 20 MHz
- Podržava jedan 10Base-T port s automatskim otkrivanjem i ispravljanjem polariteta



Slika 2.5. Izvedba ENC28J60 Ethernet Shielda na Arduino Nano pločici

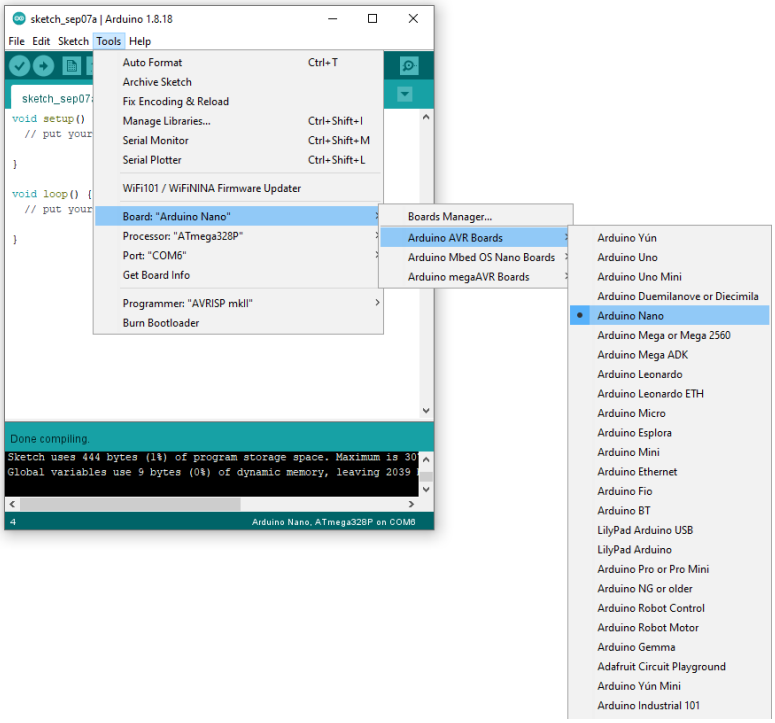
2.3. Arduino razvojno okruženje

Programski jezik koji se najčešće koristi za pisanje u Arduino software-u je C i C++, a krajnji izvršni program se prevodi u binarni kod preko prevoditelja *(eng. compailer)* . Arduino Software (IDE) je *open-source* razvojno okruženje koje nam je omogućio sam Arduino. Struktura prikazana slikom 2.6. je minimalni program Arduino software-a. Naredba setup() je početna naredba koja se izvodi kada se Arduino upali ili kada se resetira , a loop() je ustvari glavni program koji označava beskonačnu petlju. Arduino Software (IDE) pomaže pri prijenosu programskog koda na uređaj te se taj programski kod naziva „sketch“ čija datoteka ima nastavak „.ino“. Kao što vidimo na slici 2.6. sadrži i prostor za poruke koji nakon provjere programa ili spremanja prikazuje različite iznimke, greške i informacije. [9]



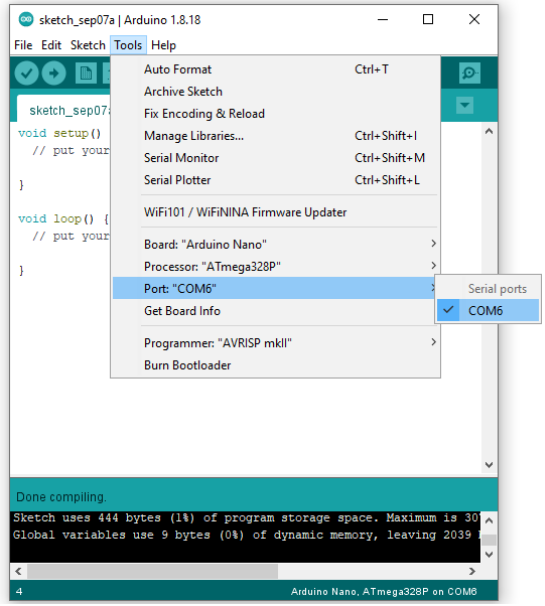
2.6. Izgled Arduino razvojnog okruženja

Kada spojimo Arduino Nano USB kablom u računalo, prvo što moramo napraviti je označiti u meni-u Alati/Ploča *(eng. Tools/Bord)* i označiti mikroupravljač kojim se koristimo; U našem slučaju Arduino Nano što pokazuje slika 2.7.

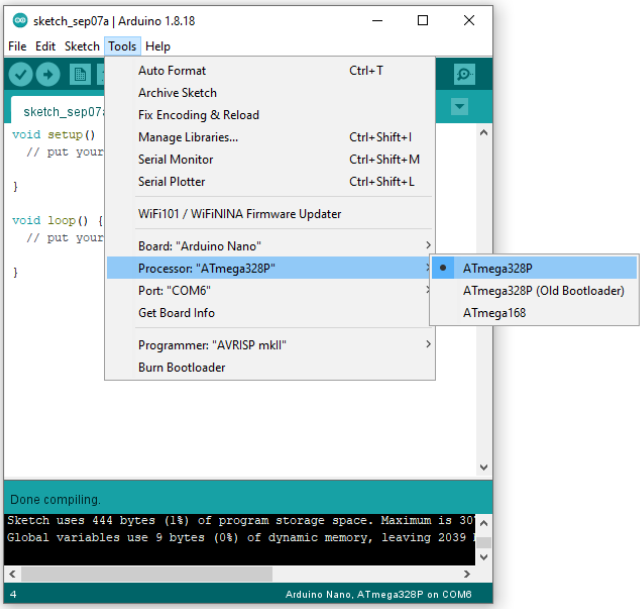


Slika 2.7. Mikroupravljač koji koristimo

Također postoje i druge postavke koje se isto nalaze u alatima, te ih moramo namjestiti, a to su ulaz (eng. Port) prema slici 2.8. i procesor (eng. Processor) prema slici 2.9. koji koristimo.

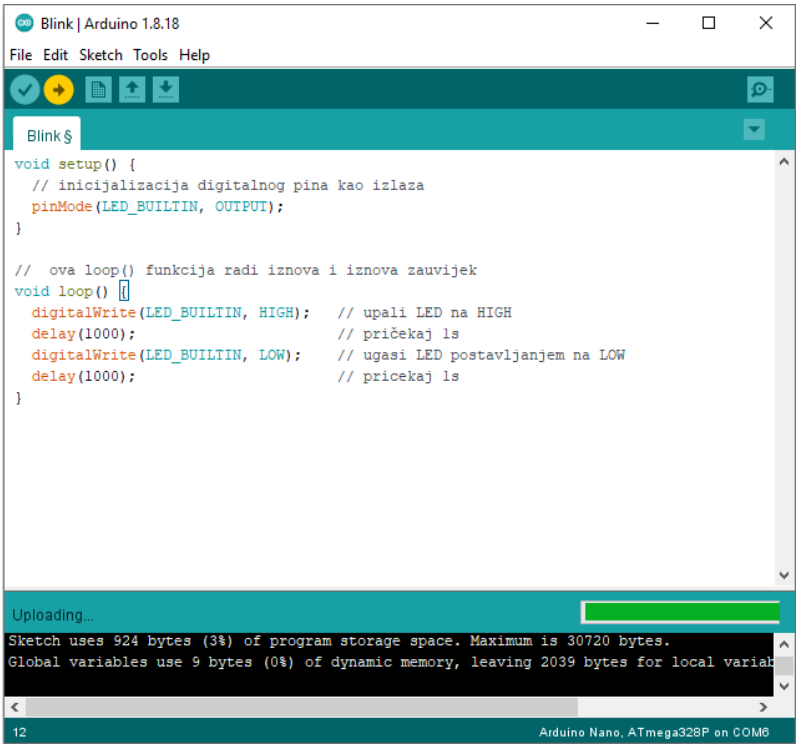


Slika 2.8. Ulaz na koji je spojen mikroupravljač USB portom



Slika 2.9. Mikročip koji koristimo

Primjer treptanja LED svijetla je najjednostavniji primjer kako bi vidjeli Arduino Software (IDE) u akciji. Ovaj primjer koristi ugrađen LED koji ima većina Arduino ploča pomoću konstante LED_BUILTIN prema slici 2.10. te nam omogućuje jednostavno upravljanje LED-om. Digitalni pin na Arduino Nano ploči za LED svijetlo je D13.



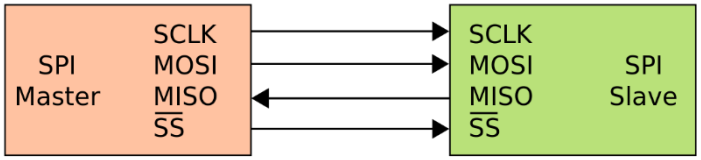
Slika 2.10. Primjer jednostavnog Arduino koda

2.4. Povezivanje i protokol

Komunikacija između elektroničkih komponenti slična je komunikaciji između ljudi. Obje komponente moraju govoriti istim jezikom kako bi se razumjele. Za elektroničke komponente ti su jezici poznati kao komunikacijski protokoli. Komunikacijski protokoli su pravila potrebna za razmjenu informacija između računala i instrumenata.

Internet se odnosi na najveću svjetsku mrežu koja povezuje velik broj uređaja diljem svijeta. Korisnicima omogućuje trenutnu razmjenu informacija. Osim toga, internet nudi mnoge resurse i usluge. Broj protokola i standarda raste iz dana u dan, a održivi su oni jednostavni i robusni. Ovo je bit IP protokola (eng. „Internet Protocol“), na kojem se temelji niz protokola više razine i koji može prenositi sve vrste podataka. Da bi uređaji mogli komunicirati putem IP protokola, potrebno je znati njihove IP adrese (adrese uređaja na mreži) i portove na kojima se poruke šalju ili primaju. IP komunicira pomoću dva protokola na transportnom sloju: TCP (eng. Transmission Control Protocol) i UDP (eng. User Datagram Protocol). TCP protokol podržava spor ali pouzdan prijenos podataka. To je protokol orijentiran na povezivanje, što znači da ako paket ne stigne do odredišta, bit će ponovno poslan. IP protokol je mreža širokog (WAN) područja dok ćemo mi koristiti lokalnu mrežu (LAN) tj. Ethernet. [10]

Arduino Ethernet Shield koristi SPI (eng. Serial Peripheral Interface) sučelje. Ono nam omogućuje komunikaciju na kratkim udaljenostima, prvenstveno u ugrađenim (eng. Embedded) sustavima. SPI uređaji komuniciraju u full duplex modu što je dvostruki komunikacijski sustav koji se koristi u mnogim komunikacijskim mrežama, a omogućuje da obje strane mogu komunicirati jedna s drugom istovremeno. Koriste master-slave arhitekturu obično s jednim masterom. Master-slave je model asimetrične komunikacije ili kontrole gdje jedan uređaj ili proces(„glavni“) kontrolira jedan ili više drugih uređaja ili procesa („podređeni“) i služi kao njihovo komunikacijsko čvorište. [7]



Slika 2.11. Osnovni primjer SPI sabirnice

2.4.1. Ethernet protokol

Ethernet (IEEE 802.3) je naziv za popularnu mrežnu tehnologiju za LAN mreže. Prvi put je stvorena 1973.godine umrežavanjem Altro računala i printera na Xerox Palo Alto istraživačkom centru od strane Bob Metcalfea. Temelji se na *frame* načinu rada što znači da se preko mreže šalju paketi koji su prilagođeni za slanje. Ethernet je sigurniji od Interneta. Ethernetu vanjski uređaji nemaju pristup mreži. Kod interneta to nije tako i to može uzrokovati više sigurnosnih prijetnji uređajima. Ethernet je postao najpopularnija LAN tehnologija, sada se pojavljuje gotovo u svim korporativnim mrežama kao i mnogim malim instalacijama.

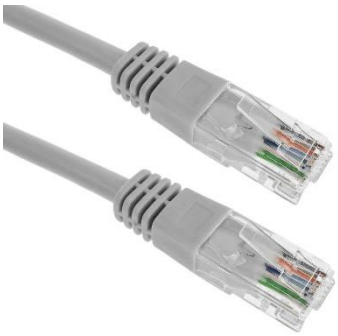
Ethernet se sastoji od tri dijela:

- UTP kabel – fizički medij kojim putuju informacije u mreži
- Protokola – pravila za kontrolu pristupa na mediju
- Ethernet paketa – u njima se prenose podatci kao skupina bitova organiziranih u polje

Da bi računala u mreži radila, sva moraju razumjeti isti protokol, odnosno sva trebaju raditi po njegovim pravilima. Ethernet protokol određuje da svaki paket završava zadanom adresom, jer prema njegovim pravilima svaki paket mora imati adresu izvora i odredišta. Svako računalo u mreži ima 48-bitni ključ koji se naziva MAC adresa, a njegova glavna zadaća je da računalima u mreži da drugu adresu. Jedinstven je za svaki proizvedeni Ethernet uređaj. Uređaj koji povezuje računalo s Ethernet mrežom naziva se mrežna kartica. U Ethernetu su svi korisnici jednaki jer nema središnjeg nadzora. Svi dijele propusnost mreže, tako da niti jedan korisnik na mreži ne može imati cijeli svoj medij. Budući da se podaci u Ethernetu šalju serijski u manjim paketima, postoji velika vjerojatnost da dva ili više korisnika istovremeno šalju neke podatke na istoj mreži. Podatci se šalju tek kada računalo provjeri medij, pa tek kad ustanovi da je slobodan počinje slat neke podatke. Ovaj kontrolni mehanizam se naziva MAC, a već smo ga spomenuli zato što je ugrađen u naš Arduino Ethernet Shield. [6]

2.5. UTP kabel

UTP kabel se uglavnom koristi za uspostavljanje Ethernet mreže te je on zamijenio ranije korišten koaksijalni kabel i AUI kabel. Ethernet oprema je vrlo jeftina jer je mrežna kartica obično već na matičnoj ploči i zahtjeva samo kabel i ako je potrebno, u slučaju neke veće mreže, mrežni preklopnik ili čvorište (*eng. switch ili eng. hub*) . Uglavnom se brzina prijenosa podataka kreće od 10MBps do 100MBps, gdje je 100MBps najčešća brzina prijenosa podataka, dok se u zadnje vrijeme najviše promovira 1GBps. [11]



Slika 2.12. UTP kabel

2.6. Switch

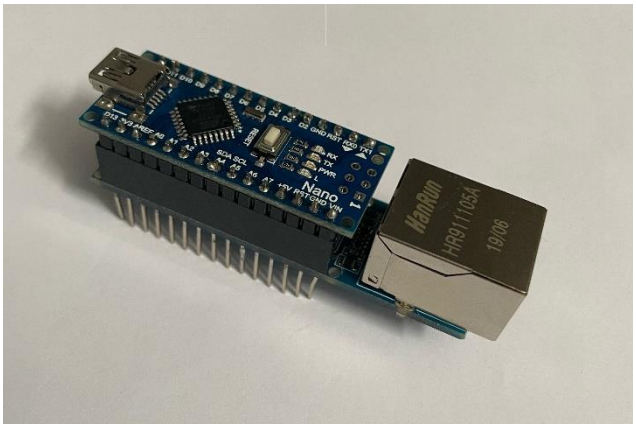
Osim UTP kabela nama je u ovom završnom radu zbog složenije mreže potreban preklopnik *(eng. Switch)*. To je uređaj koji nam služi za povezivanje tj. spajanje dva ili više računala, u našem slučaju mikroupravljača, u jednu mrežu. Njegov posao je upravljanje protoka podataka između dijelova lokalne mreže i brine se da više računala može istovremeno neometano koristiti istu mrežu. Postoje modeli sa od 2 – 24 porta te dolaze sa vanjskim napajanjem.

Slika 2.13. Switch

3. IZVEDBA KOMUNIKACIJE

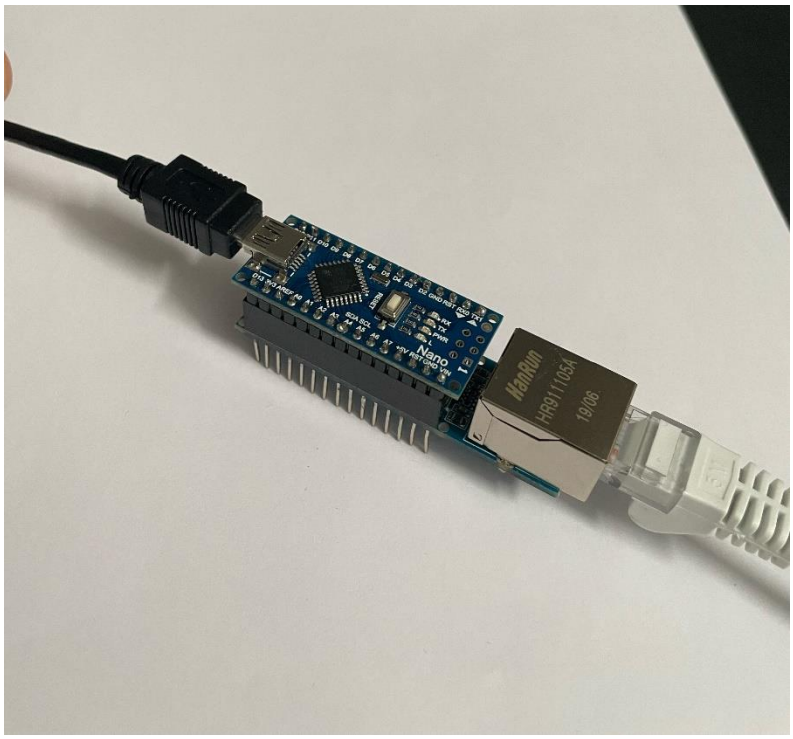
3.1. Fizička izvedba komunikacije

Prvi korak pri izvedbi komunikacije je spojiti Arduino Nano pločice sa njihovim Ethernet Shield-om.



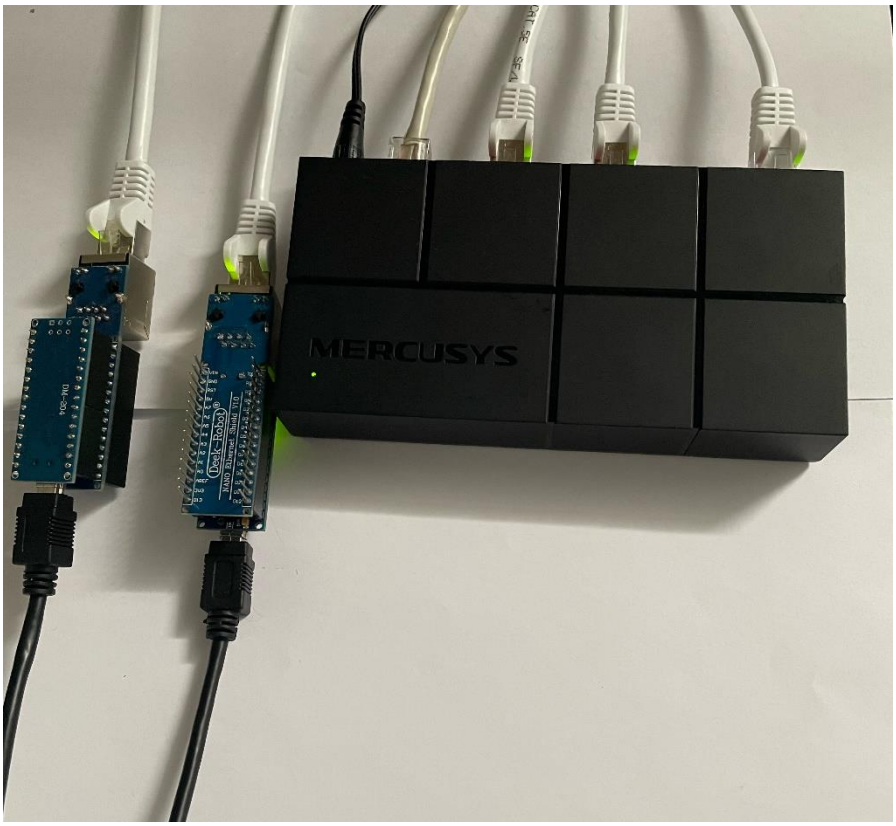
Slika 3.1. Arduino nano i ethernet shield

Drugi korak je spojiti Arduino pločicu USB kabelom na računalo, te Arduino Ethernet Shield UTP kabelom sa switch-em koji je spojen na LAN mrežu.



3.2.Završni spoj Arduino Nano pločice i Ethernet Shielda

Kada smo spojili pločice kako je prikazano na Slici 3.2. moramo ih priključit na mrežu pomoću UTP kabla i switcha koji nam je potreban kako bi mogli napraviti stabilnu LAN mrežu za više korisnika. Naravno, Arduino pločice putem USB kabla trebamo spojiti u računalo. Koristili smo 4 UTP kabela, jedan da spojimo switch na mrežu, jedan za računalo te dva za dvije Arduino pločice prema slici 3.3.



Slika 3.3. Prikaz spajanja pločica na LAN

3.2. Programska izvedba komunikacije

Nakon što smo spojili sve fizičke komponente krećemo sa radom u Arduino software-u tako što ćemo otvoriti jedan Arduino IDE prozor te Arduino pločicu 1 spojiti na port COM 5 i označiti koji mikročip koristi te otvoriti drugi prozor za Arduino pločicu 2 na COM 6 portu i također označiti korišteni čip. Napravit ćemo TCP konekciju između dvije pločice, jedna će oponašati TCP klijenta dok će druga pločica oponašati server koji će slušati poslani TCP zahtjev od klijenta. Bitove ćemo slati sa klijenta na server. Server će na svom Serial Monitor-u prikazati poslani bit(broj,znak).

Arduino #1 koji oponaša TCP klijenta aktivno šalje zahtjev za TCP vezu prema Arduino #2 što je ujedno i njegova uloga u komunikaciji. Klijenta smo konfigurirali tako što smo postavili MAC adresu njegovog Ethernet Shield-a te inicijalizirali taj shield. Pomoću server porta i lokalne IP adrese smo uspostavili konekciju . Nakon toga u petlji je omogućena komunikacija preko Serial Monitora.

```
Serial.println("ARDUINO #1: TCP CLIENT");

if (Ethernet.begin(mac) == 0)
  Serial.println("Failed to configure Ethernet using DHCP");

if (TCPclient.connect(serverAddress, serverPort))
  Serial.println("Connected to TCP server");
else
  Serial.println("Failed to connect to TCP server");
```

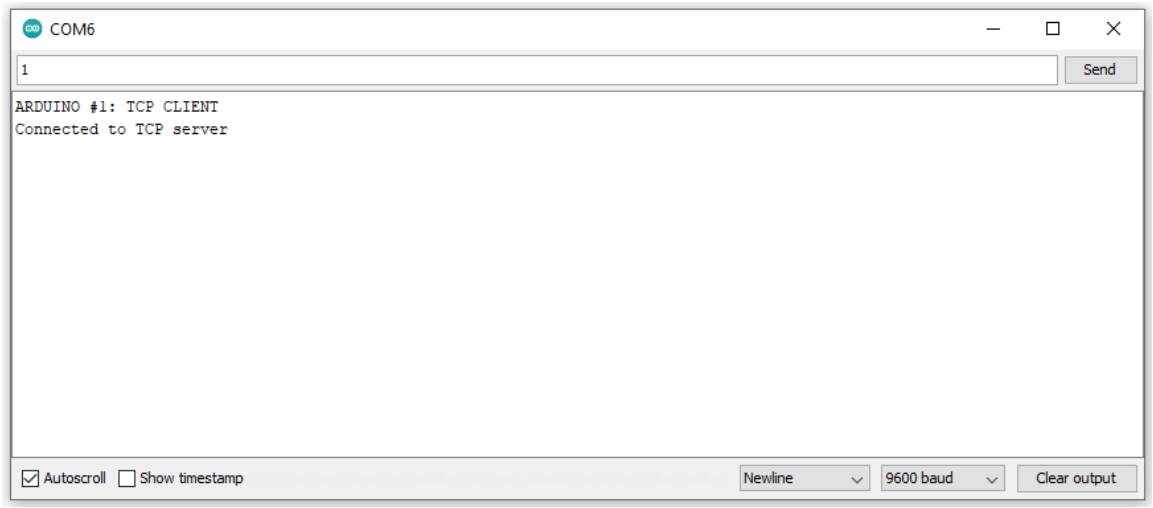
Slika 3.4. Inicijalizacija Ethernet Shielda i konekcija na TCP server

Arduino #2 koji oponaša TCP server, sluša zahtjev za povezivanje od Arduina #1 te na Serial Monitor prikazuje poslani bit. MAC adresa Arduino Ethernet Shielda od Arduino #2 mora biti različita od Arduino #1 da ne dođe do konflikta.

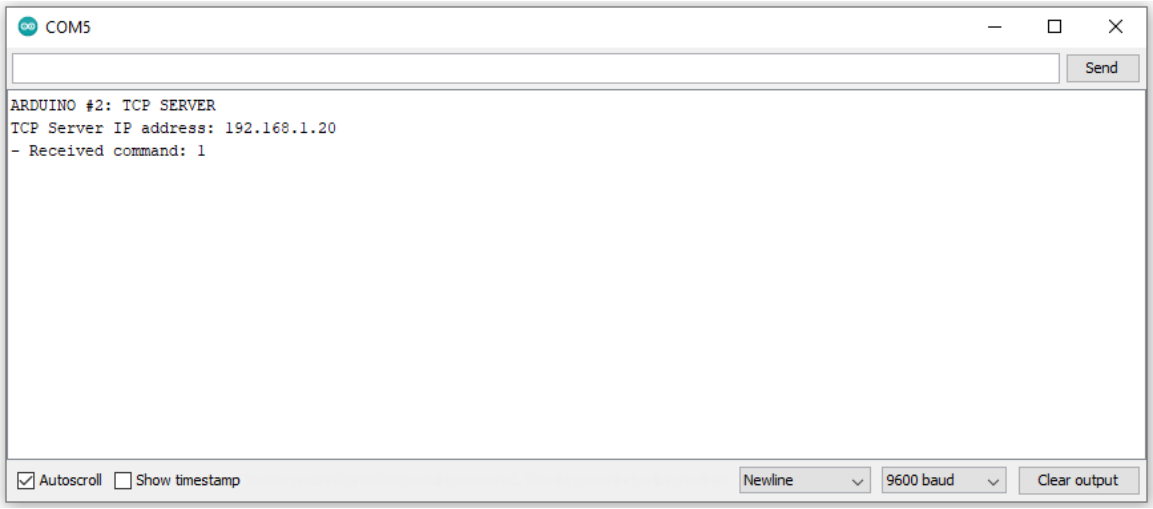
```
EthernetClient client = TCPserver.available();

if (client) {

  byte command = client.read();
  Serial.print("- Received command: ");
  Serial.write(command);
  Serial.print("\r\n");

  Ethernet.maintain();
}
```

Slika 3.5. Čitanje naredbe iz TCP klijenta

Komunikacija je vidljiva na slikama 3.6. i 3.7. , a programsko rješenje je u prilogu.



Slika 3.6. Arduino klijent Serial Monitor



Slika 3.7. Arduino server Serial Monitor

4. ZAKLJUČAK

Ovim završnim radom uspješno je ostvarena jednostavna komunikacija između dva mikroupravljača putem Ethernet protokola. Pri izradi sustava obuhvaćene tehnologije su: Ethernet, Arduino Nano, Nano Ethernet Shield, Arduino IDE . Arduino razvojno okruženje pokazalo se kao pouzdano rješenje prilikom komunikacije. Bez previše postavki Arduino Ethernet Shield je pokazao jednostavnost svoga korištenja te je njegova svrha ispunjena. Upoznali smo se sa Arduino software-om koji nam je pomogao pri realizaciji ovoga rada sa svojim karakteristikama kao što su neke od korištenih naredbi, kompatibilnost sa Arduino pločama te svojim serijskim monitorom preko kojega smo mogli vidjeti komunikaciju u akciji. Prikazali smo full-duplex prijenos podataka tj. kako funkcionira klijent-server prijenos u oba smjera, jedna radna stanica je slala podatke dok druga je druga primala podatke što se u praksi često koristi. Ovaj rad nam je zbog svoje jednostavnosti pokazao zašto je Ethernet najpopularnija LAN tehnologija i temelj većine korporativnih mreža kao i u mnogim malim instalacijama.

LITERATURA:

[1] Arduino , https://www.arduino.cc/

[2] Arduino Store, https://store.arduino.cc/products/arduino-nano

[3] Richard Electronics, richardelectronics.com/productdetail/

[4] Wikipedia, Arduino, https://hr.wikipedia.org/wiki/Arduino

[5] Wikipedia, Arduino Ethernet Shield http://wiki.sunfounder.cc/index.php?title=Ethernet_Shield

[6] Douglas, E. Comer „Internetworking with TCP / IP principles, protocols, and architectures“ , New Jersey 1995.

[7] Hammel, Bob „Connecting Arduino“ , US 2014.

[8] A Study on New Arduino NANO Board , Jordan 2020.

[9] Banzi, Massimo „Getting Started with Arduino“ , Kalifornija 2008.

[10] Wikipedia, Ethernet, https://en.wikipedia.org/wiki/Ethernet

[11] Wikipedia, Twisted pair, https://en.wikipedia.org/wiki/Twisted_pair

SAŽETAK

Naslov: Komunikacija između mikroupravljača primjenom Ethernet protokola

Ovim završnim radom je postignuta komunikacija preko LAN mreže odnosno komunikacija između mikroupravljača preko Ethernet protokola. Pomoću Arduino razvojnog okruženja je prikazana jednostavnost i korisnost dvosmjerne komunikacije putem protokola. Znanje koje smo stekli ovim završnim radom uz dodatnu nadogradnju je moguće primijeniti u većim sustavima i mrežama.

Ključne riječi: Arduino pločice, Ethernet , komunikacija, protokoli

ABSTRACT

Title: Communication between microcontrollers using the Ethernet protocol

This final work has accomplished a communication over a LAN network, that is, communication between microcontrollers via the Ethernet protocol. Using the Arduino development environment, the simplicity and usfulness of two way communication via the protocol is demonstrated. The knowledge we have gained through this final work with additional upgrades can be applied in larger systems and networks.

Keywords: Arduino boards, Ethernet, communication, protocols

ŽIVOTOPIS

Nikola Tomas rođen je 10. listopada 1996. u Osijeku. U istom gradu završava osnovnu školu „ Grigor Vitez “ , te 2011. godine upisuje Elektrotehničku i prometnu školu u Osijeku. Godine 2015. ostvaruje upis na Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, smjer računarstvo.

PRILOG A: Programsko rješenje za Arduino klijent

```
#include <SPI.h>
#include <Ethernet.h>


const int serverPort = 4080;

byte mac[] = {0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x03};
IPAddress serverAddress(192, 168, 1, 20);
EthernetClient TCPclient;

byte incomingByte;

void setup() {
  Serial.begin(9600);

  Serial.println("ARDUINO #1: TCP CLIENT");

  if (Ethernet.begin(mac) == 0)
    Serial.println("Failed to configure Ethernet using DHCP");

  if (TCPclient.connect(serverAddress, serverPort))
    Serial.println("Connected to TCP server");
  else
    Serial.println("Failed to connect to TCP server");
}

void loop() {

  if (!TCPclient.connected()) {
    Serial.println("Connection is disconnected");
    TCPclient.stop();

    if (TCPclient.connect(serverAddress, serverPort))
      Serial.println("Reconnected to TCP server");
    else
      Serial.println("Failed to reconnect to TCP server");
  }


  if(Serial.available() > 0)
  {
     incomingByte = Serial.read();

     TCPclient.write((byte)incomingByte);
     TCPclient.flush();
  }


}
```

PRILOG B: Programsko rješenje za Arduino server

```
#include <SPI.h>
#include <Ethernet.h>

const int serverPort = 4080;

byte mac[] = {0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02};
EthernetServer TCPserver(serverPort);

void setup() {
  Serial.begin(9600);

  Serial.println("ARDUINO #2: TCP SERVER ");

  if (Ethernet.begin(mac) == 0)
    Serial.println("Failed to configure Ethernet using DHCP");

  Serial.print("TCP Server IP address: ");
  Serial.println(Ethernet.localIP());

  TCPserver.begin();
}

void loop() {
  EthernetClient client = TCPserver.available();

  if (client) {

    byte command = client.read();
    Serial.print("- Received command: ");
    Serial.write(command);
    Serial.print("\r\n");

    Ethernet.maintain();
  }
}
```