

**UNIVERZITET CRNE GORE**  
**ELEKTROTEHNIČKI FAKULTET**  
**-smjer Elektronika-**



**RS485 mreža**  
**-seminarski rad iz Industrijske elektronike-**

**Autori:**

**Damjan Mustur 17/16**

**Ivan Rašović 4/16**

**Milo Janković 33/16**

**Marko Biljurić 34/16**

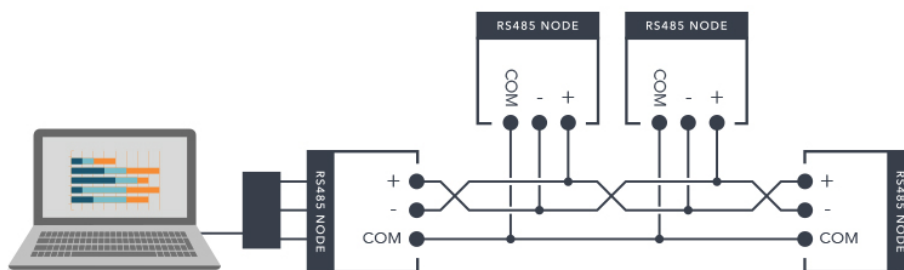
**Mentor :**

**prof. dr Radovan Stojanović**

U ovom radu ćemo predstaviti način rada RS485 mreže, i način na koji smo to realizovali. Hardverski dio smo konstruisali koristeći MAX 485 module i USB 485 konvertor a elektronsku kontrolu LED dioda i DC motora vršimo preko Arduino mikrokontrolera.

Najkorišteniji serijski metodi su RS232 i RS485. RS232 je jedan od najučestalijih protokola koji se može naći na današnjim uređajima. Modem se može konektovati sa jednim PC-em pomoću RS232, čija je maksimalna brzina prenosa 20 kb/s a domet 15m. Dva računara ne možemo povezati preko RS232 protokola. Ukoliko je potrebno napraviti mrežu više računara, senzora i aktuatora i pritom ostvarimo bolju brzinu prenosa koristi se RS485. RS485 standard je definisan od strane EIA da bi se dobio odgovor na sve nove kombinacije povezivanja. Time RS485 postaje najšire korišten serijski protokol u sistemima za preuzimanje i prenos podataka, za kontrolne aplikacije koje rade u realnom vremenu i za opštu komunikaciju između više udaljenih čvorova.

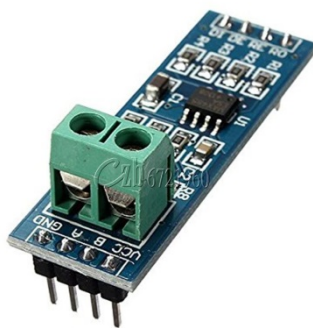
RS485 predstavlja polu-dupleks asinhronu mrežu. Moguće je slati podatke u oba smjera ali ne istovremeno. Podaci se prenose kao niz bitova.



Grafički prikaz mreže

## MAX 485 module

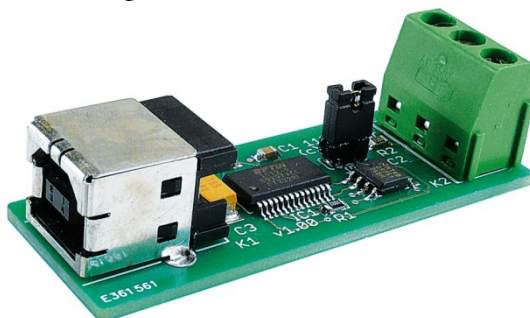
Max 485 module je primopredajnik male snage namijenjen RS-485 i RS-422 komunikacijama. Svaki dio sadrži jedan drajver i jedan prijemnik. Max 485 može da prenosi do 2Mbps. Za napajanje ovog primopredajnika je potrebno između  $120\mu\text{A}$  i  $500\mu\text{A}$  i 5V. Ovaj module ima 4 pina: po jedan za napajanje i masu i dva za signal.



Max 485 module

## USB 485 konvertor

Ovaj konvertor obezbjeđuje konverziju dvosmjernog serijskog komunikacionog signala između RS-485 i USB porta koristeći virtuelni serijski COM port.



USB 485 konvertor

## Arduino kod

Ispod je dat kod za Arduino koji je zadužen za upravljanje DC motorom. Izabrali smo L293D drajver za motor jer za svoju cijenu ima najbolje performanse i pogodan je za više vrsta motora. Komanda mora da počne sa Motor2 i taj dio stringa koji se šalje je indentifikator komande. Nakon Motor2 treba upisati cijeli broj. Taj broj predstavlja koliko sekundi treba da se motor okreće. Sve ostale komande koje šalje master će biti ignorisane od strane ovog slave-a. Dobar dodatak ovom projektu bi bila i konmanda za kontrolisanje brzine rotacije ali za to treba dodatni hardver(ESC) koji mi nismo imali u periodu realizacije ovog projekta.

Kod za pwm kontrolu LED za koju je zadužen drugi Arduino slave je dat ispod. Komanda treba da bude formata LED1 <jačina svijetlosti od 0 do 100%> .

Ovdje LED1 ima funkciju identifikatora. Sve komande koje ne počinju sa LED1 ovaj slave će ignorisati. Nakon što upali LED slave šalje nazad poruku “LED upaljen” kao znak da je zadatak uspješno odrađen.

```
#include <SoftwareSerial.h>

//define constants
#define SSerialRX    10 //RO na RS485 modulu
#define SSerialTX    11 //DI na RS485 modulu

#define SSerialTxControl  3 // DE i RE na RS485 modulu

#define RS485Transmit    HIGH
#define RS485Receive    LOW

#define LED13           13
#define A1               4
#define A2               7
#define EN1             6

SoftwareSerial RS485Serial(SSerialRX, SSerialTX); /

String stringReceived, stringSend, buff;

int outputValue, buff_int;

void setup() {
  Serial.begin(9600);
  Serial.println("Motor Slave");

  pinMode(LED13, OUTPUT);
  pinMode(A1,OUTPUT);
  pinMode(A3,OUTPUT);
  pinMode(EN1,OUTPUT);
  pinMode(SSerialTxControl, OUTPUT);

  digitalWrite(EN1,LOW);
  digitalWrite(A1,LOW);
  digitalWrite(A2,LOW);

  digitalWrite(SSerialTxControl, RS485Receive);

  RS485Serial.begin(19200);
}

void loop() {
```

```

Motor_control();
}
void Motor_control()
{
  if(RS485Serial.available())
  {
    stringReceived=RS485Serial.readString();
    digitalWrite(LED13,HIGH);
    delay(10);
    digitalWrite(LED13,LOW);
    if(stringReceived.substring(0,6)=="Motor2")
    {
      buff=stringReceived.substring(7);
      buff_int=buff.toInt();
      digitalWrite(EN1,HIGH);
      digitalWrite(A2,HIGH);
      digitalWrite(A1,LOW);
      delay(buff_int*1000);

      digitalWrite(EN1,LOW);
      digitalWrite(A1,LOW);
      digitalWrite(A2,LOW);
      digitalWrite(SSerialTxControl, RS485Transmit);
      RS485Serial.write("\nMotor uapljen\n");
      digitalWrite(SSerialTxControl,RS485Receive);
    }
    else digitalWrite(SSerialTxControl,RS485Receive);
  }
}

```

## Kod za PWM slave

```

#include <SoftwareSerial.h>

#define SSerialRX    10 // RO na RS485 modulu
#define SSerialTX    11 // DI na RS485 modulu

#define SSerialTxControl 3 //DE i RE na RS485 modulu

#define RS485Transmit  HIGH
#define RS485Receive  LOW

#define LED13        13
#define LED_PWM      9

SoftwareSerial RS485Serial(SSerialRX, SSerialTX);

String stringReceived, stringSend, buff;

int outputValue, buff_int;

void setup() {
  Serial.begin(9600);
  Serial.println("LED Slave");

  pinMode(LED13, OUTPUT);
  pinMode(LED_PWM,OUTPUT);
  pinMode(SSerialTxControl, OUTPUT);

  digitalWrite(LED_PWM,LOW);

  digitalWrite(SSerialTxControl, RS485Receive);

  RS485Serial.begin(19200);
}

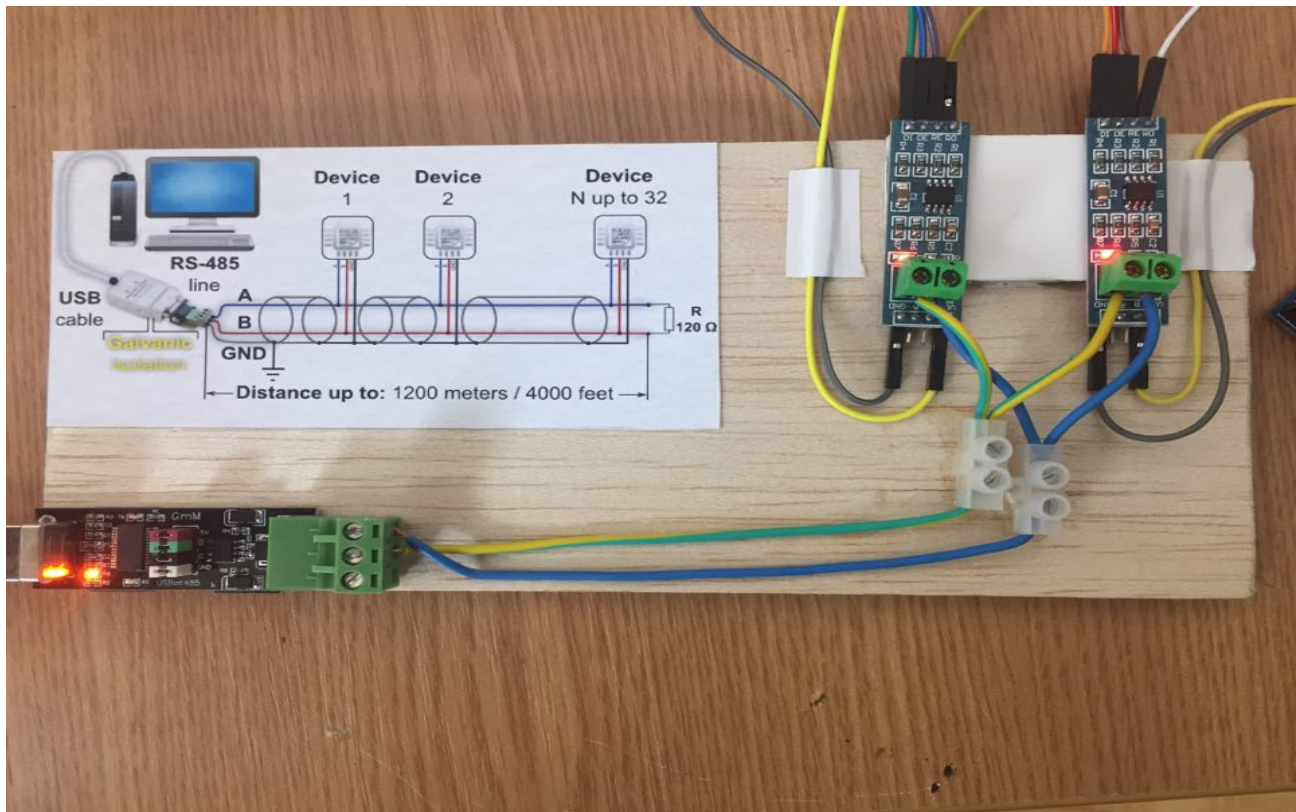
void loop() {
  Activate_LED();
}

stringReceived=RS485Serial.readString();
digitalWrite(LED13,HIGH);
delay(10);
digitalWrite(LED13,LOW);
if(stringReceived.substring(0,4)=="LED1")
{
  buff=stringReceived.substring(5);

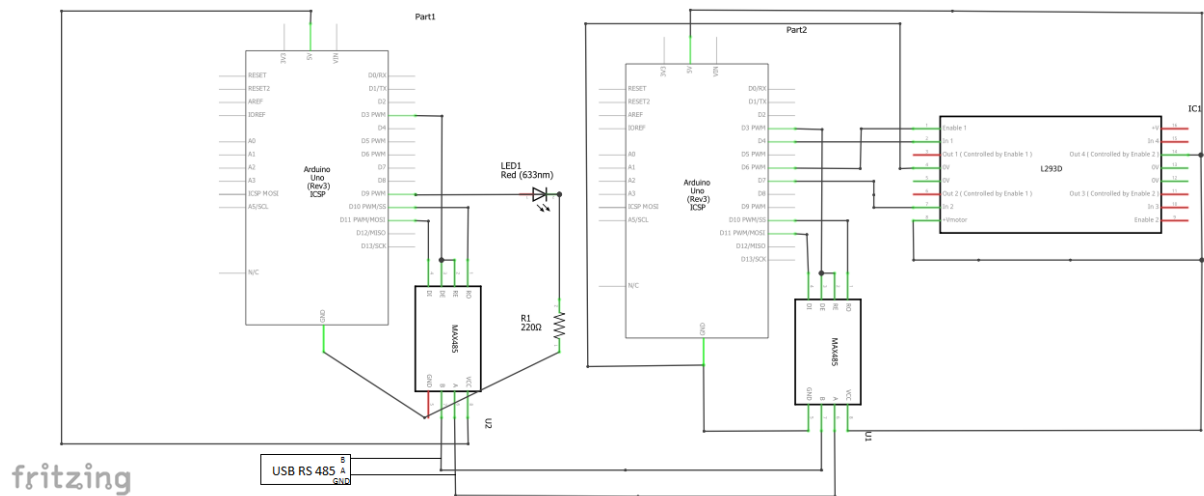
  buff_int=buff.toInt();
  outputValue = map(buff_int, 0, 100, 0, 255);
  analogWrite(LED_PWM,outputValue);
  digitalWrite(SSerialTxControl, RS485Transmit);
  RS485Serial.write("\nLED uapljen\n");
  digitalWrite(SSerialTxControl,RS485Receive);
}

}
else digitalWrite(SSerialTxControl,RS485Receive);
}

```



Maketa na kojoj je napravljen primjer RS485



Šema povezivanja za projekt

Internet Izvori :

<https://en.wikipedia.org/wiki/RS-485>

<http://www.gammon.com.au/forum/?id=11428>

<https://forum.arduino.cc/index.php?topic=406051.0>