

## Pouzdanost prenosa podataka serijske komunikacije prilikom korišćenja Uart-Usb konvertora

Nikola Cakić<sup>1,2</sup> , Aleksandar Žigić<sup>1</sup> , Srdjan Milosavljević<sup>1</sup> 

<sup>1</sup> Univerzitet u Beogradu, Elektrotehnički institut „Nikola Tesla“, Koste Glavinića 8a, 11000 Beograd, Srbija

<sup>2</sup> Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Trg Dositeja Obradovića 6, 21102 Novi Sad, Srbija

[nikola.cakic@ieent.org](mailto:nikola.cakic@ieent.org)

**Kratak sadržaj:** Prilikom dizajniranja uređaja koji se oslanja na koncept postojanja centralnog mikrokontrolera za obavljanje merenja i dodatnog računara za skladištenje i grafički prikaz rezultata, neretko se dešava da nema dovoljan broj USB interfejsa. Iz pomenutog javlja se potreba za korišćenjem određenih serijskih konvertora. U radu su prikazani rezultati testiranja pouzdanosti serijske UART-USB komunikacije prilikom korišćenja UM232R konvertora. Razvojno okruženje Tiva TM4C1294XL Launchpad rezultate merenja temperature šalje putem UART komunikacije do UM232R konvertora koji je povezan USB vezom sa ROCK PI 4 single board kompjuterom. Korisnička aplikacija se bazira na Java programskom jeziku i jSerialComm biblioteci. Komunikacija je testirana u paketima od 100, 1 000 i 10 000 nizova od po 50 karaktera. Vremenski intervali na kojima je testiran prenos su 10mS, 100mS, 500mS, 1s. Svaki test je ponovljen po 10 puta zbog validnosti podataka. Prosečni gubitak podataka je 50% dok je predloženi algoritam smanjio gubitak podataka do 30%.

**Ključne reči:** Serijski prenos, pouzdanost, gubitak podataka, USB, UART, Tiva, Rock Pi, UM232R.

## 1 Uvod

Nedostatak USB portova potrebnih za serijsku komunikaciju između određenih periferija može se nadomestiti korišćenjem UART-USB konvertora. Pomenuti konvertori koriste široko rasprostranjeni FTDI-jev čip FT232RL.

Prilikom razvoja Sistema za merenje i automatsku regulaciju protoka gasa u okviru uređaja za ispitivanje oksidacione stabilnosti transformatorskih ulja "Termoblok" [1] javila se potreba za korišćenjem pomenutih konvertora i odabran je UM232R modul za print montažu [2].

U toku procesa razvoja algoritama za obradu i prikaz podataka otkrivena su određena ograničenja prilikom korišćenja serijskog prenosa. Ograničenja se ogledaju u vidu gubitka podataka i nepostizanja deklarisanе brzine prenosa. U ovom radu su prikazani rezultati testova serijske komunikacije na različitim vremenskim intervalima i sa različitom veličinom frejma podataka.

Po definiciji UART (engl. Universal Asynchronous Receiver-Transmitter) je hardverski komunikacioni protokol koji koristi asinhronu serijsku komunikaciju konfigurabilne brzine.

Treba napomenuti da je korišćena standardna konfiguracija serijske UART veze sa parametrima 115200 baud rate, 8 data bita, bez parity bita i 1 stop bit.

U posebnoj poglavlju biće dat algoritam koji omogućava smanjenje gubitka podataka do 30%.

## 2 Testiranje serijske veze

Za testiranje komunikacije korišćen je UM232R modul koji je putem UART serijske komunikacije povezan sa razvojnim okruženjem Tiva TM4C1294XL Launchpad koje generiše rezultate merenja. Sa druge strane modul je putem USB veze vezan na single board kompjuter Rock Pi. Uz pomoć JAVA [3] korisničke aplikacije se vrši do obrada i prikaz rezultata merenja. Korisnička aplikacija koristi jSerialComm-2.9.3 [4] serijsku biblioteku.

Parametri serijske veze su 115200 baud rate, 8 data bita, bez parity i 1 stop bit. Veličina paketa koji se šalju je 50 karaktera.

Testovi pouzdanosti slanja su obavljani u paketima od po 10, 100, 1000 i 10 000 slanja na vremenskim intervalima od 10mS, 100mS i 1sec. Radi validnosti dobijenih podataka svaki test je ponavljan deset puta. Radi preglednosti podataka u tabelama će biti date prosečne vrednosti pomenutih deset testova i to korektno primljenih podataka, korigovanih paketa i ukupno nakon korekcije.

## 2.1 Testiranje komunikacije na intervalu slanja od 1s

Tabelom 1 su prikazani rezultati testova na intervalu slanja od 1sec.

*Tabela 1. Testiranje komunikacije na vremenskom intervalu od 1sec*

Broj poslanih paketa	Korektno primljenih (prosek [%])	Korektno primljenih sa algoritmom za korekciju (prosek [%])	Ukupno korektno primljenih (prosek [%])
10	54	23	77
100	51.4	24.3	75.7
1000	51.13	19.63	70.76
10000	49.92	20.13	70.05

## 2.2 Testiranje komunikacije na intervalu slanja od 100mS

Tabelom 2 su prikazani rezultati testova na intervalu slanja od 100mS.

*Tabela 2. Testiranje komunikacije na vremenskom intervalu od 100mS*

Broj poslanih paketa	Korektno primljenih (prosek [%])	Korektno primljenih sa algoritmom za korekciju (prosek [%])	Ukupno korektno primljenih (prosek [%])
10	53	23	78
100	51.8	24.1	75.9
1000	50.47	20.05	70.52
10000	50.66	21.23	71.89

Pored naglaska na korektno primljene podatke prilikom slanja na 100mS otkriven je gubitak podataka od 10% između poslanih i primljenih podataka.

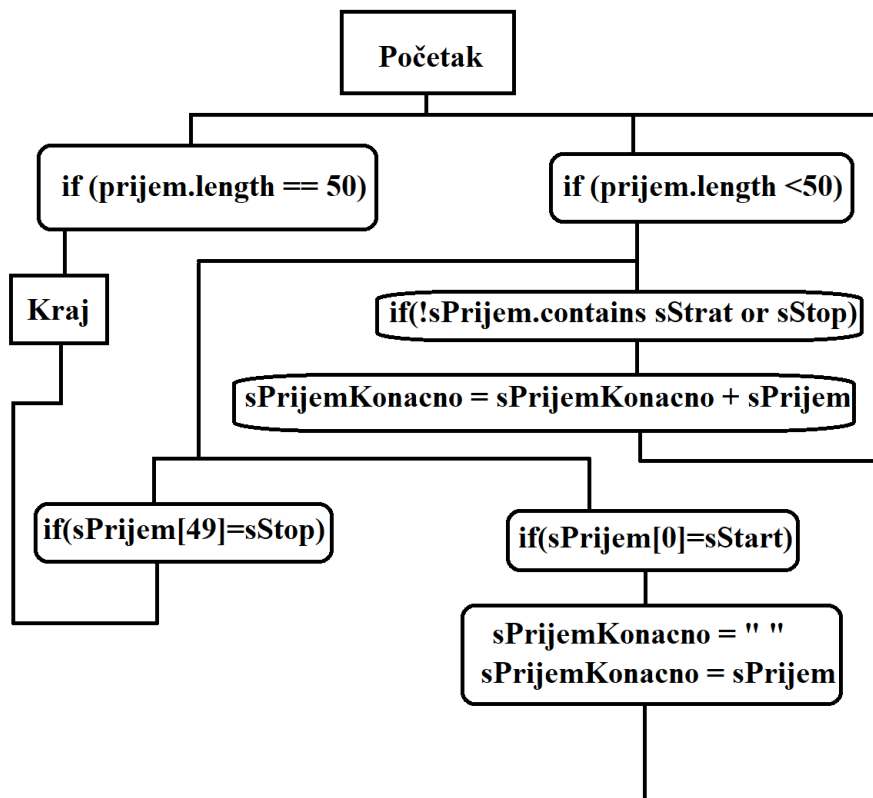
## 2.3 Testiranje komunikacije na intervalu slanja od 10mS

Samo testiranje na 100mS otkrilo je da je dostignuta najveće brzina komunikacije. Prilikom testova na 10mS potvrđeno je samo da je maksimalna

brzina slanja podataka 111mS jer se broj primljenih podatak smanjio toliko da više nije moguće staviti naglasak na procenat korektno primljenih podataka.

### 3 Predloženi algoritam za povećanje broja korektno primljenih paketa

Ono što je ustanovljeno prilikom testova je da se dešava da jedan paket bude korektno primljen ali ne iz prijema jednog paketa nego iz više uzastopnih paketa. Slikom 1 je dat predloženi algoritam da bi se povećao broj korektno primljenih paketa.



Slika 1. Predloženi algoritam

Algoritam radi sledeće ako se prijem sastoji od 50 karaktera detektovan je prijem poruke. Poruka se onda testira da li sadrži simbole za početak i kraj poruke.

Ako je pristigla poruka manja od 50 karaktera ispituje se da li sadrži simbol za start poruke. Ukoliko sadrži string sPrijemKonacno se obriše i upiše se u njega primljena poruka sPrijem. Zatim se testira da li je stigla cela poruka ako jeste ona se uzima u obzir i sPrijemKonacno se opet briše, ako ne nego je sledeća poruka isto manja od 50 karaktera onda se proverava da li poruka sadrži simbol za kraj poruke. Ukoliko poruka sadrži simbol za kraj poruke testira se da li kada se sabere veličina poruke u kojoj je detektovan simbol za početak poruke sa porukom koja sadrži simbol za kraj poruke ima ukupno 50 karaktera. Ukoliko ima 50 karaktera detektovan je prijem poruke.

Ukoliko je nakon prijema paketa koji sadrži simbol za kraj poruke pristigla poruka koja ima manje od 50 karaktera a u sebi ne sadrži simbol za kraj poruke sPrijemKonačno se inkrementira za sPrijem sve do 50 karaktera gde se daje mogućnost da se detektuje simbol za kraj poruke i potencijalni prijem korektne poruke

## 4 Zaključak

Nakon svih testova nameću se dva zaključka da UART-USB kovertorski modul ima ograničenja u brzini prilikom slanja podataka na 111mS po jednom slanju i da se dešava problem da poslati paket može stići na stranu prijema u vidu više paketa.

Predloženim algoritmom se može povećati procenat korektno primljenih podataka za do 30%. Ono što je interesantno da se nikad nije desilo da je poslat ni jedan bajt više od poslatih bajtova ali se dešavalo da jedan paket bude korektno poslat ali iz više paketa.

Ako se uzme u obzir podešavanje UART komunikacije od 115200 bit/s i da je niz karaktera 50 karaktera sledi da je za jedno slanje potrebno  $50 \cdot (8+1)$  što je 450 bita. Sledi da je teoretski moguće poslati 256 poruka u sekundi ili na svakih oko 4mS, što nažalost nije slučaj u realnim uslovima.

## Literatura

- [1] Aleksandar Žigić, Nenad Kartalović, Nikola Cakić, Jelena Lukić, Srđan Milosavljević, "Sistem za merenje i automatsku regulaciju protoka gasa u okviru uređaja za ispitivanje oksidacione stabilnosti transformatorskih ulja "Termoblok"", *Zbornik radova, Elektrotehnički instituta „Nikola Tesla“*, Vol. 24, 2014.

- [2] UM232R USB Serial UART Development Module Datasheet, [https://ftdichip.com/wp-content/uploads/2020/08/DS\\_UM232R.pdf](https://ftdichip.com/wp-content/uploads/2020/08/DS_UM232R.pdf)
- [3] Java (programming language), [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- [4] **jSerialComm**, <https://fazecast.github.io/jSerialComm/>

**Abstract.** During the design of the device that is based on a central microcontroller for measurements and an additional computer for graphical presentation and storage of measurement results, it often happens that there are not enough available USB ports. Therefore the need for using particular serial converters arises. This paper presents the results of reliability testing of serial UART-USB communication using UM232R converter. The development environment Tiva TM4C1294XL microcontroller Launchpad sends temperature measurement results using UART communication to UM232R converter which is connected to ROCK PI 4 single board computer via USB interface. The user application is based on Java programming language and jSerialComm library. The communication is tested using packets of 100, 1 000 i 10 000 arrays of 50 characters. Time intervals used for tests are 10mS, 100mS, 500mS, and 1s. Each test is repeated 10 times to improve test validity. The average data loss is 50%, while the proposed algorithm reduced the loss up to 30%.

**Keywords:** Serial transfer, Reliability, Data loss, USB, UART, Tiva, Rock Pi, UM232R.

## **The Reliability of Serial Data Communication Using Uart-Usb converter**

Nikola Cakić, Aleksandar Žigić, Srdjan Milosavljević

Rad primljen u uredništvo: 09.12.2022. godine.

Rad prihvaćen: 15.12.2022. godine.