

Arduino uCFTP server i bežična komunikacija sa WPF grafičkom aplikacijom *

Sadržaj

1	Zadatak projekta	1
1.1	Prethodni projekat	1
1.2	Zadatak	1
2	Mentor	2
2.1	Reference	2
3	Izgled klijentske aplikacije	2
4	Opis klijentskog koda	2
4.1	Struktura projekta	2
4.2	MainWindow.xaml.cs	3
4.3	EditorPrinter.cs	3
4.4	ClientSender.cs	4

1 Zadatak projekta

1.1 Prethodni projekat

Prethodni projekat i njegovu dokumentaciju možemo naći na adresi:
<https://github.com/stefanpns/ESP32ArduinoUCFtpSystem>

1.2 Zadatak

Zadatak ovog projekta bio je dopunjavanje prethodnog projekta. Klijentska aplikacija sada sadrži grafički interfejs koji klijent može da koristi za slanje komande serveru i primanje odgovora pomoću *HTTP POST* poruka. Koristili smo *Windows Presentation Foundation* (WPF) softverski okvir za izradu grafičke aplikacije.

*uCFtp predstavlja kvazi-FTP protokol za mikrokontrolere

2 Mentor

Izrada projekta bila je u sklopu predmeta *Seminarski D* i mentor na ovom projektu bio je izabrani profesor Đorđe Herceg sa Prirodno-matematičkog fakulteta u Novom Sadu departman za matematiku.

2.1 Reference

Većinu programskog koda koji koristimo na serverskoj strani predstavljaju kodovi koje je napisao profesor Đorđe Herceg.

Originalne kodove možemo naći na adresi: <https://github.com/djherceg>

3 Izgled klijentske aplikacije

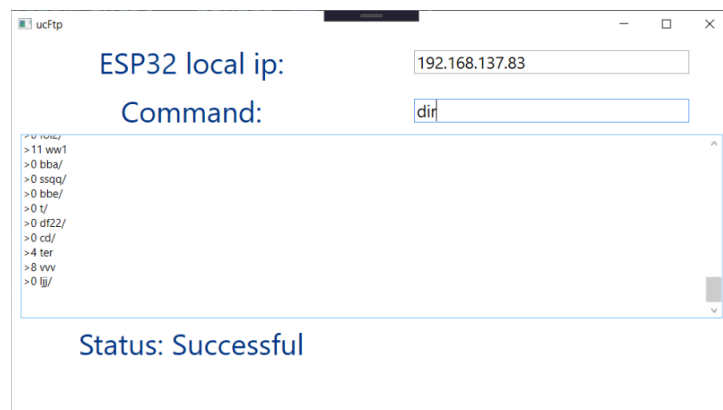


Figure 1: Izgled klijentske aplikacije

4 Opis klijentskog koda

4.1 Struktura projekta

1. U folderu *ucFTP* sadržane su već postojeće klase iz prethodnog projekta. Opis klasa i njihovih upotreba možemo naći u dokumentaciji prethodnog projekta (1.1.)
2. Sledeće klase predstavljaju nove klase:
 - (a) *MainWindow.xaml.cs* (pogledati podsekciju 4.2.)
 - (b) *EditorPrinter.cs* (pogledati podsekciju 4.3.)
 - (c) *ClientSender.cs* (pogledati podsekciju 4.4.)

4.2 MainWindow.xaml.cs

Opis

U fajlu *MainWindow.xaml.cs* nalazi se klasa *MainWindow* pomoću koje definišemo i upravljamo grafičkim elementima. Definisali smo sledeća polja:

1. polje za unos ip lokacije servera
2. polje za unos komande
3. polje za ispis izvršenih komandi
4. polje za ispis stanja aplikacije. Koristimo polje za ispis statusa slanja i primanja *HTTP* zahteva i odgovora.

Definisali smo sledeće upravljače događajima:

1. Za polje za unos komande - nakon unosa `Enter` ključa unutar polja za unos komande:
 - (a) čitamo vrednost koja se nalazi u polju za unos ip lokacije servera
 - (b) čitamo vrednost koja se nalazi u polju za unos komande
 - (c) šaljemo komandu pomoću metode
`sendCommand(string commandToSend, string ipAdress)`
objekta klase *ClientSender*. Klasu *ClientSender* objasnićemo kasnije u podsekciji 4.4.

4.3 EditorPrinter.cs

Opis

U fajlu *EditorPrinter.cs* nalazi se klasa *EditorPrinter* pomoću koje ispisujemo:

1. rezultat izvršenih komandi
2. status slanja *HTTP* zahteva
3. status primanja *HTTP* odgovora

Sadrži sledeće metode:

- `setup(TextBox resultBox, Label statusLabel)`
Pomoću ove metode inicijalizujemo reference na sledeće grafičke elemente:
 1. polje u kojem ispisujemo rezultate izvršenih komandi,
`TextBox resultBox`
 2. polje u kojem ispisujemo status slanja i primanja *HTTP* zahteva i odgovora,
`Label statusLabel`
- `PrintStatus(string status)`
Pomoću ove metode menjamo sadržaj status polja.
- `WriteLine(), WriteLine(string text), Write(string text)`
Pomoću ovih metoda dodajemo sadržaj polju rezultata izvršenih komandi.
- `scrollToEndResultBox()`
Pomoću ove metode automatski prikazujemo zadnji red u polju koje sadrži rezultate izvršenih komandi.

4.4 ClientSender.cs

Opis

U fajlu *ClientSender.cs* nalazi se klasa *ClientSender* pomoću koje šaljemo *HTTP POST* zahtev serveru. Sadrži sledeće metode:

- `sendCommand(string commandToSend, string ipAdress)`
Pomoću ove metode šaljemo komandu (`commandToSend`) serveru. Server se nalazi na adresi `ipAdress`. Ova metoda je slično struktuirana kao i metoda sa nazivom *Main* klase *Program.cs* iz prethodnog projekta (1.1.) Jedina razlika predstavlja način ispisa rezultata izvršene komande i statusa slanja i primanja *HTTP* zahteva i odgovora. Ispis sada vršimo u grafičkim elementima. Ispis vršimo pomoću klase *EditorPrinter* (4.3.)