

Piotr Nowicki
Informatyka, niestacjonarne, sem. IX
nr indeksu: 112715

Sprawozdanie z Podstaw Teleinformatyki

Informatyka, studnia niestacjonarne, sem. VII

Repozytorium: <https://github.com/lajder/CastScreenServer>

Zawartość

I.	Opis zadania.	2
II.	Założenia realizacyjne.....	2
III.	Napotkane problemy i ograniczenia.	2
IV.	Opis działania aplikacji.	2
V.	Użytkowanie aplikacji.	3
	Wymagania wstępne dla Windows:	3
	Wymagania wstępne dla odbiorców:	3
	Uruchomienie aplikacji.	3
	Tryb bez parametrów:	3
	Tryb z parametrami:	5
	Wyświetlanie ekranu.	5
	Kończenie działania.	6

I. Opis zadania.

Stworzenie aplikacji przesyłającej zawartość ekranu na wiele monitorów innych komputerów podłączonych do tej samej sieci. Możliwość skonfigurowania i uruchomienia aplikacji w trybie automatycznym dla jednostki bez własnego monitora.

II. Założenia realizacyjne.

Aplikacja działa na systemie Windows 10 z zainstalowanym frameworkiem .Net Core 3.1 i udostępnia obraz ekranu na innych jednostkach posiadających przeglądarkę HTTP z obsługą JavaScript (dowolny system operacyjny) przez serwer HTTP.

Komunikacja pomiędzy jednostkami odbywa się przy użyciu sieci TCP/IP w obrębie tej samej sieci z wykorzystaniem protokołu HTTP.

III. Napotkane problemy i ograniczenia.

Początkowo implementacja zakładała wykorzystanie wbudowanego mechanizmu systemu Windows 10 do projekcji obrazu na urządzenia obsługujące WiFi Display. Niestety API Windows w tej technologii nie do końca obsługuje aplikacje desktopowe WinForms, a wykorzystanie UWP wiązało się z wieloma ograniczeniami i zdawało się za bardzo powielać dostępną funkcjonalność systemu operacyjnego.

Kolejnym ograniczeniem systemu Windows okazała się możliwość utworzenia pulpitu wirtualnego, a raczej konieczność zaimplementowania sterownika do rozszerzenia pulpitu, co było poza zasięgiem przy założonych ramach czasowych. Rozwiązaniem może okazać się zainstalowanie i wykorzystanie sterownika komercyjnego i wykorzystanie tej aplikacji do przesłania rozszerzonego pulpitu na inne jednostki.

Do poprawnego działania aplikacji niezbędna jest odpowiednia konfiguracja systemu Windows oraz sieci wewnętrznej. Należy odblokować port na których jednostki będą się komunikować, a sama aplikacja, jako że korzysta z HTTPListener biblioteki System.Net musi posiadać w systemie odpowiedni wpis z adresem na którym zostanie uruchomiony serwer HTTP dla aktualnego użytkownika (lub grupy użytkowników) lub musi zostać uruchomiona z uprawnieniami administratora.

IV. Opis działania aplikacji.

Rozwiązanie składa się z jednej aplikacji uruchamianej na systemie Windows. Zastosowano podejście w którym implementacja zależy jedynie od jednostki z której obraz będzie udostępniany i nie ma potrzeby pisania osobnego klienta do odbioru danych. Powinien wystarczyć dowolny system operacyjny z zainstalowaną przeglądarką HTTP z obsługą JavaScript podłączony do tej samej sieci. Aplikację można uruchomić w dwóch trybach: bez parametrów oraz z parametrami. W trybie bez parametrów aplikacja najpierw uruchamia interfejs graficzny w którym umożliwia wybranie interfejsu sieciowego oraz portu na którym będzie pracować serwer HTTP, następnie należy wybrać zatwierdzić przyciskiem „Start Cast”.

W trybie z parametrami istnieje możliwość uruchomienia aplikacji z konsoli (np. korzystając zdalnie z SSH, kiedy stacja nie ma podłączonego ekranu) lub skryptu, podając dwa parametry startowe: adres

IP interfejsu sieciowego oraz port. W trybie tym, jeżeli parametry są poprawne, nastąpi uruchomienie aplikacji i automatyczne rozpoczęcie nadawania obrazu. Aplikacja w osobnym wątku uruchamia mechanizm tworzenia zrzutów ekranu głównego, rysowania kursora myszki i zapisywania w odpowiednich odstępach czasowych do pamięci operacyjnej. W kolejnym wątku uruchamiany jest serwer HTTP z mechanizmem nasłuchiwanie żądań od klientów zewnętrznych. W odpowiedzi, dostarczany jest najpierw plik HTML i CSS z definicją przestrzeni do wyświetlania ekranu i skryptem odświeżania obrazu. Następnie na żądanie przeglądarki z uruchomionym skryptem JavaScript przesyłany jest bieżący zrzut ekranu w odpowiednim formacie. Aplikacja tworzy zrzuty ekranu i odpowiada na żądania HTTP do póki nie zostanie wciśnięty przycisk „Stop Cast”, bądź do momentu wyłączenia aplikacji.

V. Użytkowanie aplikacji.

Wymagania wstępne dla Windows:

- Zainstalowany framework .Net Core 3.1,
- zasada firewall pozwalająca na komunikację na odpowiednim porcie
- uprawnienia administratora lub dodany wpis do znanych adresów HTTP dla danego użytkownika (polecenie: „netsh http add urlacl url=http://+:80/ user=DOMAIN\user”),
- połączenie do wewnętrznej LAN

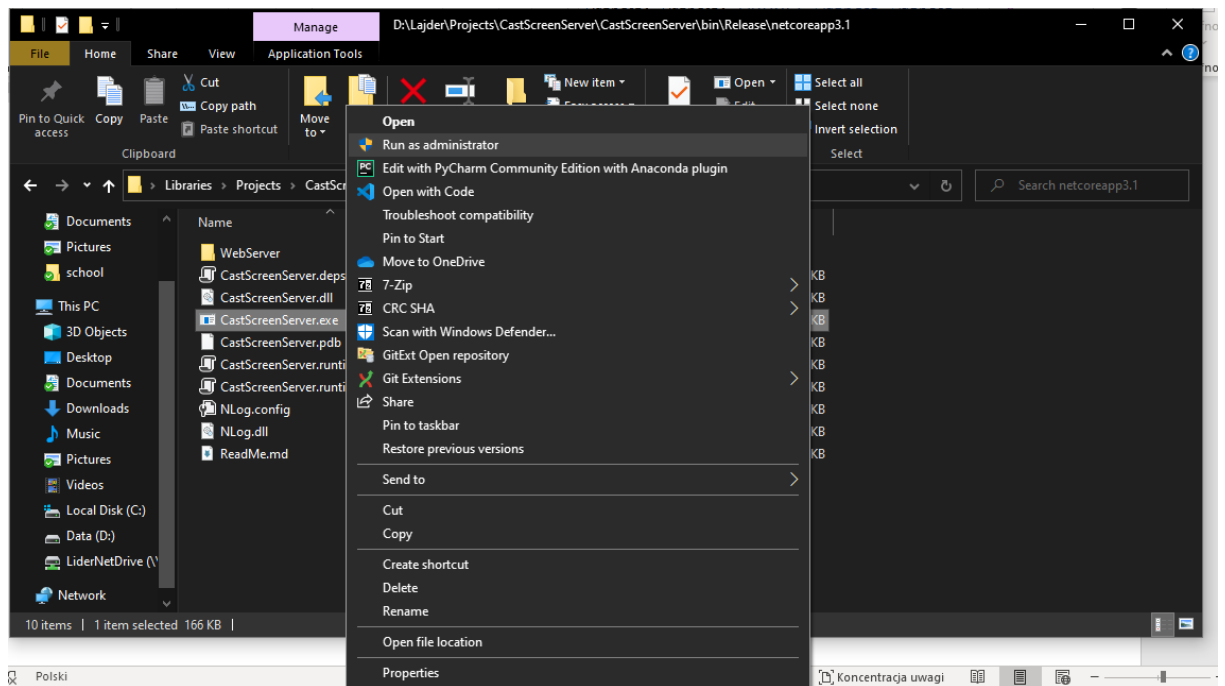
Wymagania wstępne dla odbiorców:

- Przeglądarka HTTP z obsługą HTML5 i JavaScript (rekomendowana: FireFox 73.0.1 lub Microsoft Edge 44.18362.449.0),
- połączenie do wewnętrznej LAN.

Uruchomienie aplikacji.

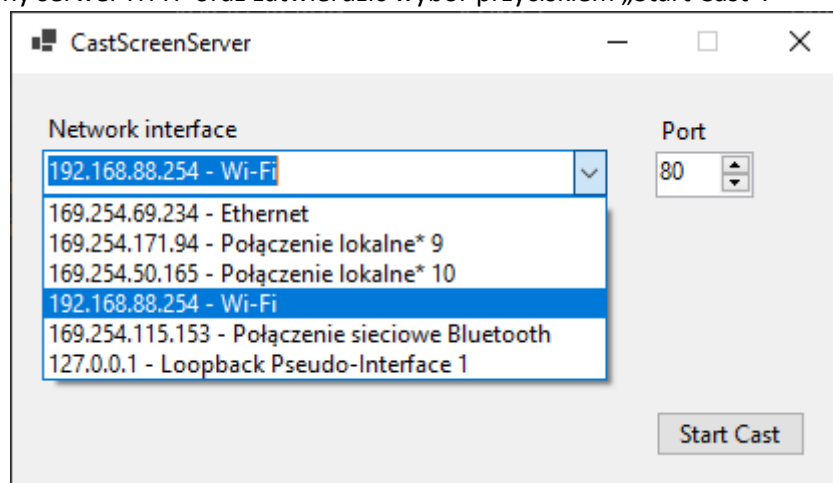
Tryb bez parametrów:

1. Jeżeli został wcześniej dodany wpis dla bieżącego użytkownika z adresem serwera HTTP, to aplikację można uruchomić tak jak każdą inną aplikację desktopową. Jeżeli możliwy jest dostęp do uprawnień administratora, można uruchomić aplikację z podwyższonymi uprawnieniami. [Rysunek 1.]



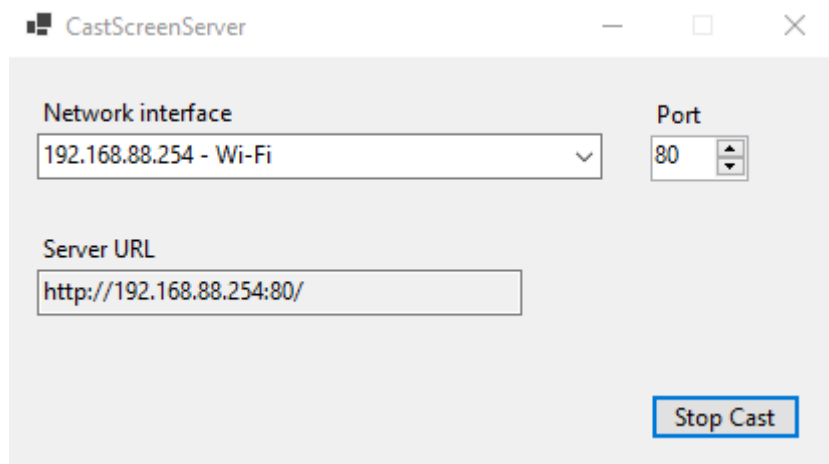
Rysunek 1 Uruchomienie bez parametrów

- Po uruchomieniu aplikacji należy wybrać interfejs sieciowy oraz port na którym zostanie uruchomiony serwer HTTP oraz zatwierdzić wybór przyciskiem „Start Cast”.



Rysunek 2 Wybór interfejsu sieciowego

- Aplikacja rozpocznie przechwytywanie ekranu oraz uruchomi serwer HTTP ze wskazanym w odpowiednim polu adresem URL. [Rysunek 3.]



Rysunek 3 Przekazywanie obrazu

Tryb z parametrami:

Podczas uruchamiania aplikacji możliwe jest przekazanie parametrów z predefiniowanym adresem IP interfejsu sieciowego oraz numerem portu, dzięki czemu udostępnianie obrazu rozpocznie się automatycznie. Komenda: „CastScreenServer.exe \$ipAddr \$Port”

```
Administrator: Command Prompt
D:\Lajder\Projects\CastScreenServer\CastScreenServer\bin\Release\netcoreapp3.1>dir
Volume in drive D is Data
Volume Serial Number is E0B5-E32D

Directory of D:\Lajder\Projects\CastScreenServer\CastScreenServer\bin\Release\netcoreapp3.1

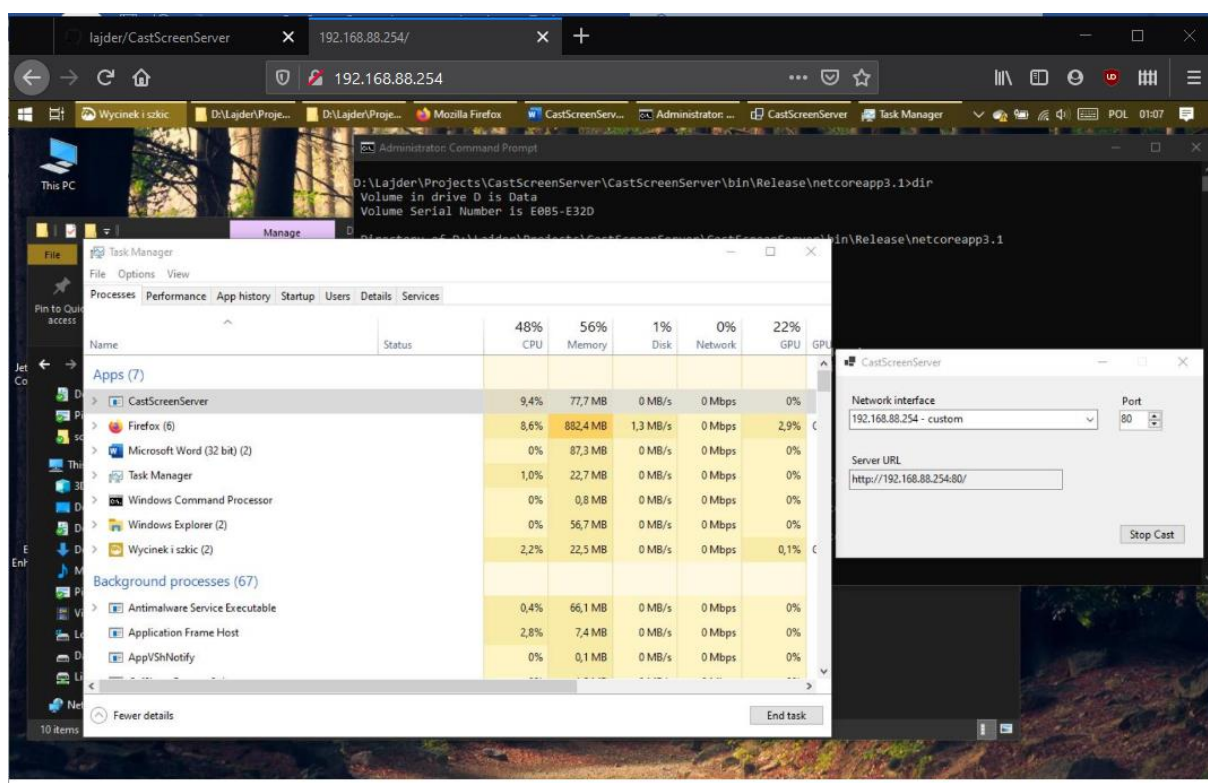
2020-02-19 19:04 <DIR>          .
2020-02-19 19:04 <DIR>          ..
2020-02-19 19:04             1 754 CastScreenServer.deps.json
2020-02-19 19:04             16 896 CastScreenServer.dll
2020-02-19 19:04            169 984 CastScreenServer.exe
2020-02-19 19:04              3 352 CastScreenServer.pdb
2020-02-19 19:04             278 CastScreenServer.runtimeconfig.dev.json
2020-02-19 19:04             161 CastScreenServer.runtimeconfig.json
2020-02-19 19:04              1 368 NLog.config
2020-02-19 19:04             781 824 NLog.dll
2020-02-19 19:04             179 ReadMe.md
2020-02-19 19:04 <DIR>          WebServer
                9 File(s)          975 796 bytes
                3 Dir(s)  31 208 906 752 bytes free

D:\Lajder\Projects\CastScreenServer\CastScreenServer\bin\Release\netcoreapp3.1>CastScreenServer.exe 192.168.88.254 80
```

Rysunek 4 Uruchomienie aplikacji z parametrami

Wyświetlanie ekranu.

Po poprawnym uruchomieniu aplikacji i rozpoczęciu transmitowania zawartości ekranu przez aplikację można wyświetlić ją na wielu innych urządzeniach w sieci. W tym celu na stanowiskach docelowych należy uruchomić przeglądarkę HTTP w polu adresu wpisać adres IP i port na którym nasłuchuje serwer HTTP. [Rysunek 5.]



Rysunek 5 Okno przeglądarki z podglądem na ekran źródłowy

Kończenie działania.

W celu wstrzymania transmisji ekranu na inne jednostki należy wcisnąć przycisk „Stop Cast”. Aplikacja przestaje wtedy tworzyć nowe zrzuty ekranu oraz nie odpowiada na żądania HTML. Transmisję można wznowić wciskając przycisk „Start Cast”. W celu całkowitego zakończenia działania programu należy zamknąć okno aplikacji.