I. Treść zadania.

Warstwa łącza danych (*data link layer*) w sieciach komunikacyjnych definiuje między innymi ramkę komunikacyjną. W sieci MODBUS RTU ramka komunikacyjna pozwala na przekazywanie rozkazów lub zapytań jednostki nadrzędnej i na udzielanie odpowiedzi przez jednostki podporządkowane. Ramka komunikacyjna składa się z pól. Każde pole składa się z bajtów. Można zatem powiedzieć, że ramka jest uporządkowanym ciągiem tworzących pola. Należy założyć, Że kolejne bajty ramki przesyłane są jeden po drugim bez zbędnej zwłoki czasowej. W rzeczywistości tak może nie być. Specyfikacja sieci MODBUS narzuca pewne wymagania na maksymalny dopuszczalny czas zwłoki pomiędzy kolejnymi bajtami ramki, który nie narusza jej ciągłości. Jest on równy półtorakrotnemu czasowi trwania transmisji jednego bajtu. Każdy bajt jest wysyłany w postaci 10 lub 11 bitowej sekwencji zwanej znakiem. Znak składa się z: bitu startu, 8 bitów danych, 0 lub 1 bitu kontroli parzystości i 1 bitu stopu. Znaki wysyłane są w ramce asynchronicznie. Jednak w obrębie znaku, kolejne bity wysyłane są synchronicznie. Jeśli np. prędkość transmisji wynosi 115200b/s, to nominalny czas trwania transmisji jednego bitu wynosi 8,68μs. Ponieważ do transmisji 1 bajtu informacji potrzeba użycia 10 bitów (transfer znaków bez bitu parzystości) lub 11 bitów (transfer znaków z bitem parzystości), to czas trwania transmisji 1 znaku wynosi odpowiednio: 86,84μs lub 95,48μs. Zatem maksymalny dopuszczalny okres czasu pomiędzy kolejnymi bajtami ramki nie naruszający jej integralności nie może być dłuższy niż 130,2 lub143,23μs. Czas ten jest oczywiście tym dłuższy im niższa jest prędkość transmisji.

Zadanie 4 polega na sprawdzeniu na ile warunek na maksymalny dopuszczalny okres czasu pomiędzy kolejnymi bajtami ramki jest realizowalny w przypadku realizacji programu emulującego (do pewnego stopnia) jednostkę nadrzędną na komputerze klasy PC z systemem operacyjnym Windows xx. Zakłada się, że edycja i wersja systemu operacyjnego Windows, który będzie wykorzystany do realizacji zadania 4 jest w gestii osoby wykonującej projekt.

Specyfikacja sieci MODBUS nie obejmuje definicji jednej warstwy fizycznej. Najbardziej popularny sposób jej realizacji polega na zastosowaniu naprzemiennej komunikacji szeregowej zgodnej ze standardem RS-485. Standard sprzętowy RS-485 nie jest i nie był stosowany w standardowych komputerach klasy PC. W komputerach tych stosowany był standard asynchronicznej komunikacji szeregowej jednoczesnej RS-232. Obecnie standard ten został wyparty przez standard komunikacji USB. Do fizycznej realizacji jednostki typu master przy użyciu komputera klasy PC stosowane są odpowiednie konwertery: RS-232 → RS-485 lub USB → RS-485. Do realizacji **zadania 4** konwertery nie są potrzebne. Emulator jednostki nadrzędnej będzie jednak musiał się odwołać do rzeczywistego lub wirtualnego portu transmisji szeregowej.

Zadanie 4.1 (za 4 punkty)
Cele zadania: Określenie przydatności systemu rodziny Windows np. Windows XP do realizacji zadania komunikacyjnego w czasie rzeczywistym. Poszukiwanie rozwiązań praktycznych.

Treść zadania:

a) Należy napisać aplikację wysyłającą w sposób ciągły na dowolny port szeregowy jeden bajt np. o kodzie heksadecymalnym *0xC4*. Należy przyjąć następujące parametry transmisji: 115200, 8,n,1. Należy dokonać pomiaru odstępów czasowych pomiędzy wysyłaniem kolejnych bajtów. W tym celu można np. zastosować metodę wyznaczania średniego czasu transmisji jednego bajtu w bloku n-bajtowym. Następnie należy sporządzić wykres częstości empirycznych (histogram) czasów transmisji jednego znaku. Uwaga, przed uruchomieniem programu należy zamknąć wszystkie aktywne aplikacje.

b) Wyznaczenie rozkładu czasu transmisji należy powtórzyć przy aktywnych 10 aplikacjach. Czy średni czas transmisji 1 znaku ulegnie zmianie?

c) Zbadać wpływ operacji na oknach (np. ręczne przeciąganie i skalowanie okien) na średni czas transmisji 1 znaku.

Zadanie 4.2 (za 6 punktów)
Cele zadania: Określenie przydatności systemu rodziny Windows np. Windows XP do realizacji zadania komunikacyjnego w czasie rzeczywistym. Poszukiwanie rozwiązań praktycznych.

Treść zadania:
Należy napisać aplikację wysyłającą w sposób ciągły na dowolny port szeregowy następującą sekwencję bajtów w zapisie heksadecymalnym: 01 03 0000 0002 C40B. Sekwencja ta może być interpretowana jako zapytanie jednostki nadrzędnej o zawartość dwóch rejestrów RTU (o adresach 0 i 1) jednostki podporządkowanej o adresie 1. Parametry transmisji: 115200,8,n,1. Oczywiście w tym przypadku, bajty w czasie transmisji zostaną zamienione na 10 bitowe znaki.

a) Należy dokonać pomiaru rozkładu czasów odstępu pomiędzy kolejnymi znakami w ramce. W tym celu można ew. zastępczo zastosować metodę wyznaczania średniego czasu transmisji jednej ramki. Z tego czasu można oszacować średni odstęp pomiędzy znakami w ramce. Następnie należy sporządzić wykres częstości empirycznych (histogram) czasów transmisji jednego znaku.

b) Wyznaczenie rozkładu czasów odstępów należy powtórzyć przy aktywnych 10 aplikacjach.

c) Zbadać wpływ operacji na oknach (np. ręczne przeciąganie i skalowanie okien) na średni czas odstępu pomiędzy znakami.

Forma zaliczenia zadania projektowego
Sprawozdanie z jego realizacji zawierające odpowiednie wykresy i wnioski.

Uwagi
1) Proszę przestudiować zagadnienie wywłaszczalności zadań.
2) Zadanie może być realizowane także pod dowolnym innym systemem operacyjnym.

II. Rozwiązanie.

Zadania zostały rozwiązane na komputerze typu laptop z systemem Windows 7 Professional SP1 64-bit, o następującej konfiguracji:

Procesor:	Intel(R) Core(TM) i5-4300M CPU @ 2.60GHz 2.60 GHz
Zainstalowana pamięć (RAM):	8,00 GB (dostępne: 7,71 GB)

Program emulujący został zrealizowany w języku C wykorzystując WinAPI do obsługi wirtualnego portu COM, środowisko Code::Blocks. Do stworzenia wirtualnego portu COM został wykorzystany program: „Virtual Serial Port Driver 9.0”.

Ustawiono następujące parametry transmisji (w obu przypadkach):

```
dcb.BaudRate = CBR_115200;
dcb.fParity = TRUE;
dcb.Parity = NOPARITY;
dcb.StopBits = ONESTOPBIT;
dcb.ByteSize = 8;
```

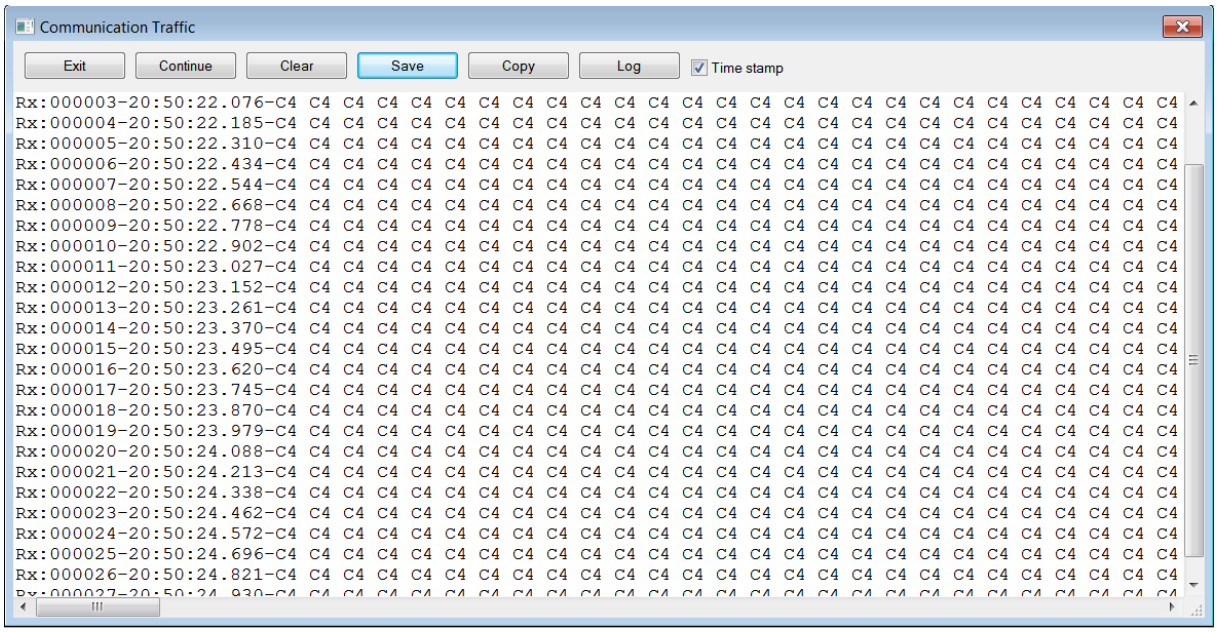
Dodatkowo w celu sprawdzenia poprawności transmisji został wykorzystany gotowy emulator jednostki slave: „Modbus Slave”. Pozwala on na symulację rejestrów jednostki slave, a także na podgląd bajtów które ogóle przychodzą do jednostki.

W celu wyznaczenia czasu transmisji jednego znaku, wykorzystano metodę wyznaczania średniego czasu transmisji jednego bajtu w bloku n-bajtowym. Było to spowodowane uzyskiwaniem niskich wartości czasów podczas pomiaru czasu wysłania pojedynczego bajta.

Zadanie 4.1

Na wirtualny port szeregowy była wysyłana wiadomość zawierająca 200-bajtowy blok o wartościach 0xC4. Taka wiadomość została wysłana 200 razy. Do pomiaru czasu została wykorzystana funkcja clock().

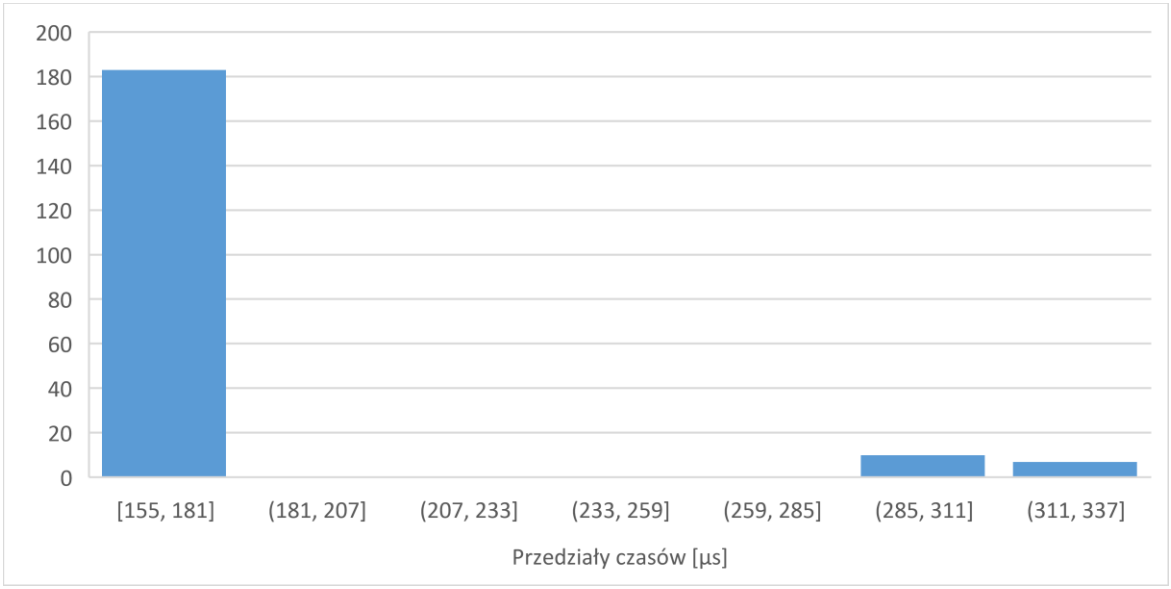
Sprawdzono poprawność przychodzących danych za pomocą programu „Modbus Slave”. Poniżej przedstawiono podgląd transmisji:



Rysunek 1 – Podgląd transmisji bajtu 0xC4 za pomocą programu Modbus Slave

a) Wyznaczenie średnich czasów transmisji przy wyłączonych aplikacjach.

Poniżej przedstawiono uzyskane wyniki w postaci histogramu (kolejne słupki oznaczają przedziały średnich czasów, w jakich został wysłany jeden znak):

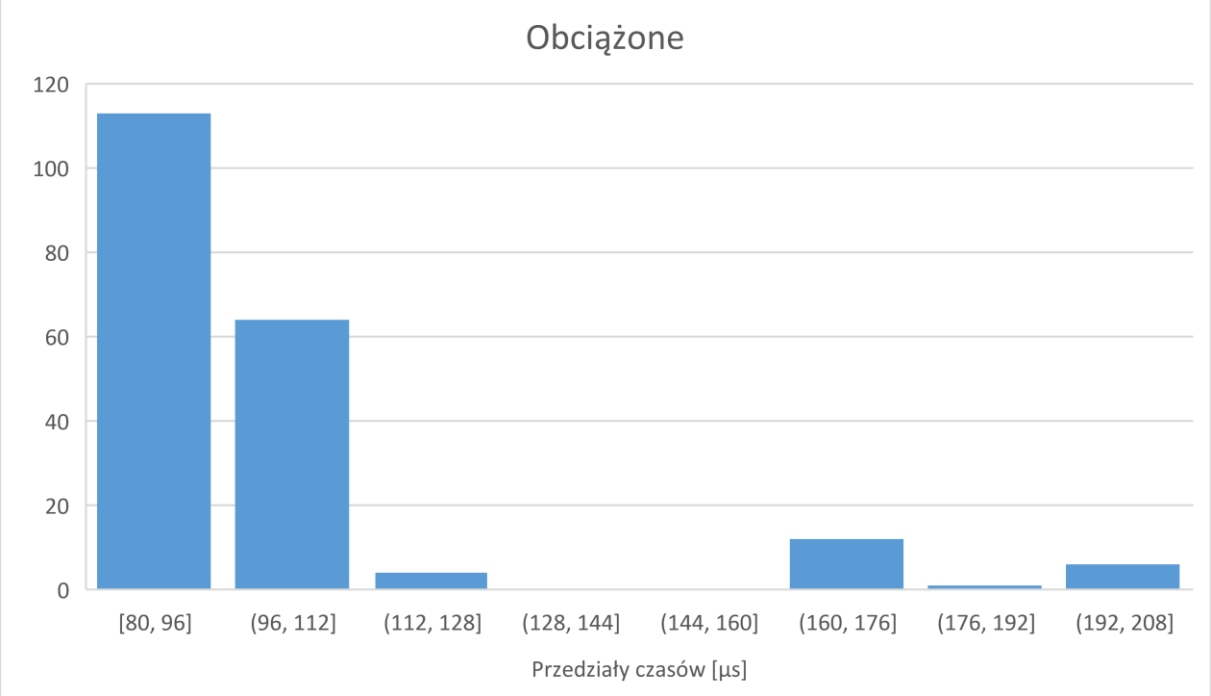


Rysunek 2 – Histogram czasów transmisji 1 bajtu, bez obciążenia

Jak widać, uzyskane czasy w większości są w zakresie 155-181 µs (co jest też zakresem najmniejszych czasów), co jest wartością naruszającą integralność ramki typu Modbus RTU. Jednak nie ma pewności, jakie procesy w tle mogły być włączone podczas testu (zostały wyłączone tylko widoczne aplikacje), więc mogło to rzutować na uzyskane wyniki.

b) Wyznaczenie średnich czasów transmisji przy obciążeniu.

W celu obciążenia uruchomiono 10 aktywnych aplikacji (np. Excel, Word, Firefox, Chrome, GIMP, PowerPoint itp.). Otrzymane wyniki przedstawiono poniżej:



Rysunek 3 – Histogram czasów transmisji 1 bajtu, obciążone

Warto zwrócić uwagę, że średnie czasy uległy zmniejszeniu, w porównaniu do poprzedniego przypadku. Ciężko określić co jest tego przyczyną. Możliwe jest, że bez obciążenia komputer/procesor jest w trybie oszczędzania energii (coś w stylu: gdy nie jest wymagana moc obliczeniowa, obniż taktowanie) i nie pracuje z pełną mocą. Natomiast gdy zostanie obciążony, to zasoby sprzętowe są wykorzystywane w pełni. Należy też zastanowić się, czy taki stan rzeczy nie wpływał na wyniku uzyskane w poprzednim punkcie. Może przez takie zachowanie systemu, sposób w jaki był mierzony czas nie był poprawny.

Mimo wszystko około 20 bajtów zostało wysłanych w czasie większym niż 130,2 µs, co nie jest dopuszczalne w sieci Modbus RTU. Wniosek z tego płynie taki, że system Windows 7 nie jest systemem czasu rzeczywistego i przydziela „różną moc" do obsługi portu COM, w zależności od stanu systemu.

c) Zbadanie wpływu operacji na oknach na średni czas transmisji 1 znaku.

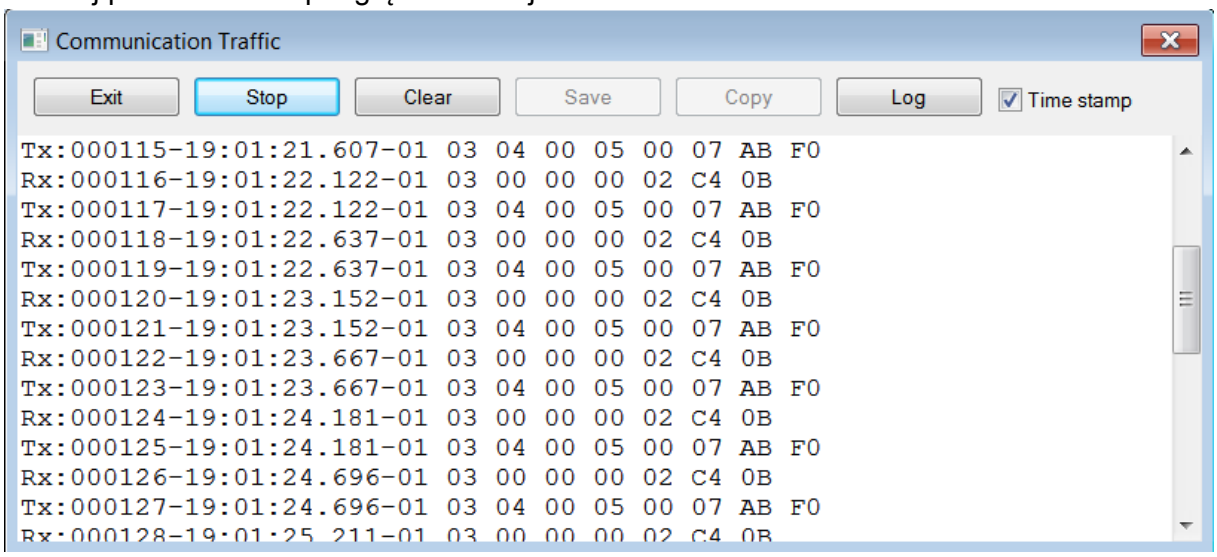
Podczas testu uruchomiono dodatkowe aplikacje jak w poprzednim punkcie i podczas transmisji skalowano okna. Nie zaobserwowano znaczących zmian w porównaniu do poprzedniego przypadku. Mogło być tak, że inny rdzeń procesora odpowiadał za transmisję, a inny za operację na oknach. Jednak takie operacje z pewnością wpływają na czas transmisji (system reaguje na skalowanie, więc w tym samym czasie transmisja powinna zostać w pewien sposób ograniczona).

Zadanie 4.2

W tym przypadku na wirtualny port szeregowy będzie wysyłana następująca sekwencja bajtów: 01 03 00 00 00 02 C4 0B. Każdy bajt w czasie transmisji zostaje zamieniony na 10 bitowe znaki. Wartość średnia pomiędzy bajtami została obliczona w następujący sposób:

- pomiar czasu wysłania 100 sekwencji bajtów: czas,
- obliczenie średniego czasu transmisji 1 sekwencji: czas/100,
- oszacowanie średniego odstępu pomiędzy znakami: czas/100/8

Sprawdzono poprawność przychodzących danych za pomocą programu „Modbus Slave". Poniżej przedstawiono podgląd transmisji:



Rysunek 4 – Podgląd transmisji wiadomości za pomocą programu Modbus Slave

Widać też odpowiedź emulowanej jednostki Slave, (rejestr 0 = „0x05", rejestr 1 = „0x07")'

a) Wyznaczenie średnich czasów transmisji przy wyłączonych aplikacjach.

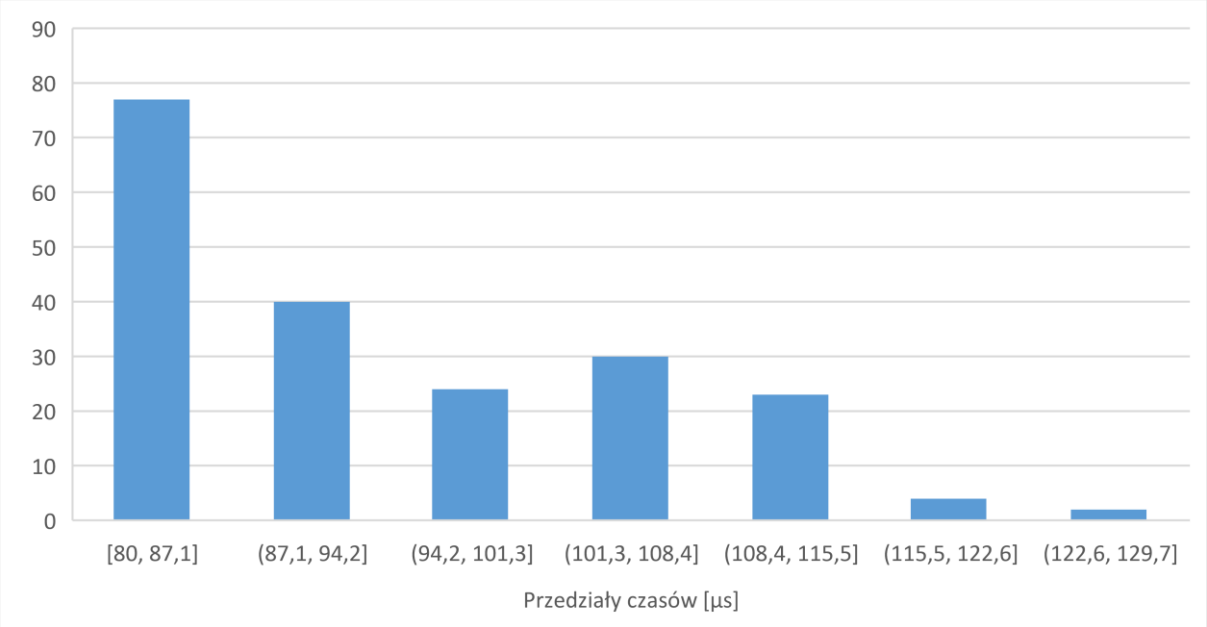
Poniżej przedstawiono uzyskane wyniki w postaci histogramu:

Rysunek 5 – Histogram czasów między znakami, bez obciążenia

Jak widać, uzyskane czasy w większości są w zakresie 155-166 µs, ale dużo czasów jest też w zakresie 188-199 µs. Odstępy między ramkami nie są więc zdeterminowane i są za duże dla sieci Modbus RTU.

b) Wyznaczenie średnich czasów transmisji przy obciążeniu.

W celu obciążenia uruchomiono 10 aktywnych aplikacji (np. Excel, Word, Firefox, Chrome, GIMP, PowerPoint itp.). Otrzymane wyniki przedstawiono poniżej:



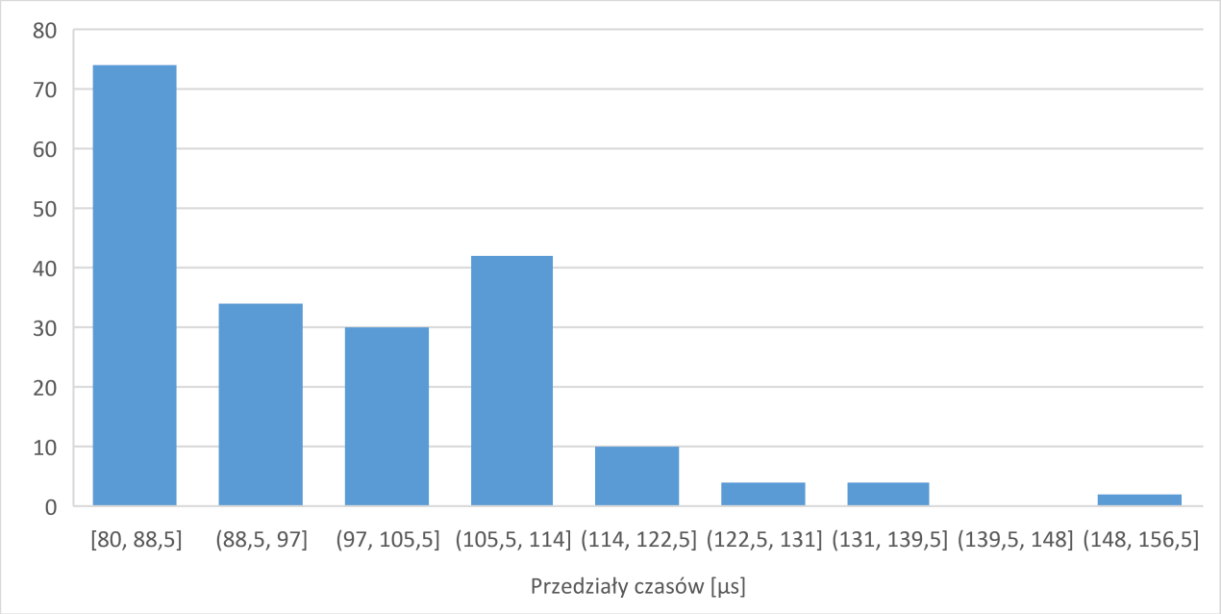
Rysunek 6 – Histogram czasów między znakami, obciążone

Średnie czasy między znakami uległy zmniejszeniu, w porównaniu do poprzedniego przypadku (tak jak w zadaniu 4.1

Co ciekawe, żaden bajt nie został wysłanyw czasie większym niż 130,2 µs. Prawdopodobnie takie wiadomości zostałyby zinterpretowane poprawnie w sieci Modbus RTU (należy jednak pamiętać, że są to średnie czasy, i niektóre bajty mogły przekroczyć tą wartość).

c) Zbadanie wpływu operacji na oknach na średni czas transmisji 1 znaku.

Podczas transmisji obciążonej dodatkowymi aplikacjami, dokonywano operacji na oknach. Otrzymane wyniki przedstawiono poniżej:



Rysunek 7 – Histogram czasów między znakami, obciążone z operacjami na oknach

Można zauważyć, że średnie czasy między znakami uległy zwiększeniu w porównaniu do poprzedniego przypadku. Niektóre przekroczyły nawet wartość 130,2 µs.

III. Wnioski.

- System Windows (w wersji która jest dołączana do komputerów osobistych) nie jest systemem czasu rzeczywistego. Wynika to z sposobu harmonogramowania i ustawiania priorytetów procesów (ten system Windows jest systemem wielozadaniowym – musi odpowiednio przydzielać zasoby do procesów). Podczas testów, aplikacja wykonująca transmisję danych na port szeregowy miała przydzielony niski priorytet, więc transmisja mogła zostać opóźnione przez inne procesy. Wynika to z mechanizmu wywłaszczenia, który polega na wstrzymaniu aktualnie wykonywanego zadania, aby inny mógł się wykonać. Wywłaszczanie umożliwia określenie przez jaki czas proces ma dostęp do zasobów procesora, co umożliwia (z punktu widzenia użytkownika) w miarę nieprzerwaną i stabilną pracę systemu i aplikacji (np. ciągłe wykonywanie się jakiejś aplikacji pochłaniającej całe zasoby sprzętowe, nie zablokuje możliwości korzystania z innej).
- W wykorzystanym komputerze istnieją procesy systemowe, których nie da się wyłączyć (są zbyt istotne dla systemu) i będą miały większy priorytet niż testowania aplikacja.
- W użytkowanym komputerze istnieją „mechanizmy” mające na celu oszczędzanie energii. Jest bardzo prawdopodobne, że wpływa to znacząco na uzyskane wyniki.
- Istnieją odmiany systemu Windows: „Windows Embedded” (lub nowa nazwa: „Windows IoT”), które mają stanowić rodzinę systemów czasu rzeczywistego. Prawdopodobnie mogą one dać oczekiwane rezultaty w przypadku sieci Modbus RTU.