

IK-401 – dostęp do kanałów szeregowych z kilku aplikacji

1. Jak zachowuje się IK-401 jeżeli dwie lub kilka aplikacji od strony LAN lub WAN w protokole TCP próbuje odczytać ten sam przelicznik na tym samym kanale szeregowym ?
2. Jak zachowuje się IK-401 jeżeli dwie aplikacje od strony LAN lub WAN w protokole TCP próbuje odczytać ten sam przelicznik na różnych kanałach szeregowych ?
3. Jak zachowuje się IK-401 jeżeli dwie lub kilka aplikacji od strony LAN lub WAN w protokole UDP próbuje odczytać ten sam przelicznik na tym samym kanale szeregowym ?
4. Czy ten sam przelicznik można odczytać zarówno w protokole TCP jak i w UDP ?
5. Czy po nawiązaniu połączenia z przelicznikiem na porcie szeregowym są w IK-401 blokowane przekierowania z WAN do LAN ?
6. Takie sprawdzenia zostały wykonane w Plum

Ad pkt 1.

Po nawiązaniu połączenia TCP z IK-401 przez jedną z aplikacji dostęp z innych aplikacji jest przez IK-401 zablokowany do czasu zamknięcia bieżącego połączenia. Tylko aplikacja z uzgodnionego podczas nawiązywania połączenia adresu źródłowego IP i portu źródłowego ma dostęp do przelicznika. Zablokowanie dotyczy innych adresów źródłowych i innych portów źródłowych. Nie ma więc znaczenia to, czy inne aplikacje próbują nawiązać połączenie od strony LAN czy od strony WAN. Nie ma znaczenia to, czy inne aplikacje są kolejnymi instancjami tego samego programu na tym samym komputerze (mają wtedy inne porty źródłowe) czy są to inne aplikacje na tym samym lub innym komputerze.

Aplikacje powinny wielokrotnie próbować nawiązywać połączenie z IK-401, aby po zamknięciu bieżącego połączenia uzyskać w końcu dostęp do przelicznika.

W kanale szeregowym do którego podłączony jest przelicznik powinno się wybrać raczej większą prędkość transmisji, aby skrócić czas bieżącego połączenia.

Aplikacja po nawiązaniu połączenia, odczytaniu przelicznika, powinna zamknąć bieżące połączenie TCP aby pozwolić nawiązać połączenie z innej aplikacji.

W IK-401, w kategorii „RS485”, czas „Timeout” w [sek] powinno się wybierać raczej krótszy (np. 8...15sek) niż dłuższy, aby IK-401 jak najszybciej zamknęła bieżące połączenie, w którym nie ma przepływu danych.

Ad pkt 2.

Po uruchomieniu obu aplikacji transmisja TCP przebiega „równoległe partiami”. Inaczej mówiąc oba kanały nie blokują się nawzajem na czas nawiązanego połączenia tylko wstrzymują się nawzajem na „krótkie” chwile.

Ad pkt 3.

Uwaga 1. Uruchomione na tym samym komputerze dwie instancje tej samej aplikacji PlumCONF v0.164 WEB nie nadają się do przeprowadzenia tego testu z powodu tego, że lokalny (źródłowy) port UDP w tej aplikacji jest nieedytowalny tylko stały (=9000). Dwie instancje mają w tym przypadku jednakowy źródłowy adres IP (komputera) oraz jednakowy źródłowy port (=9000). Nie jest to normalny przypadek w Windows i powoduje „zawieszenie” transmisji UDP. A więc test transmisji UDP z dwóch aplikacji PlumCONF v0.164 WEB można przeprowadzić wyłącznie z dwóch komputerów.

Uwaga 2. Aplikacje PlumREADER oraz GasWin2014 mają zaimplementowany protokół UDP.

Uwaga 3. Program ConfIT!_v1.0.6.10_2019-09-30 nie ma jeszcze zaimplementowanej transmisji w protokole UDP.

Uwaga 4. Aplikacja PlumCONF v0.164 WEB ma niewątpliwą zaletę względem pozostałych aplikacji taką, że na bieżąco prezentuje postęp wczytywania całej tablicy DP (stosunkowo dużo danych przez dłuższy czas). W teście IK-401, w którym chcemy sprawdzić czy i jak IK-401 blokuje dostęp innym aplikacjom jest to podstawowa zaleta. Pozostałe wymienione aplikacje ukazują jedynie momenty początku i końca wczytywania danych.

Transmisja z dwóch różnych aplikacji uruchomionych na tym samym komputerze do tego samego przelicznika nie jest wykorzystywana w praktyce a więc dalej zostanie pominięta.

W teście dwie aplikacje PlumCONF v0.164 WEB znajdują się na dwóch różnych komputerach.

Po odebraniu zapytania UDP do przelicznika od jednej z aplikacji, IK-401 zapamiętuje adres źródłowy i port źródłowy tej aplikacji oraz blokuje dostęp innym aplikacjom, o innym adresie IP i innym porcie.

Ponieważ UDP jest protokołem bezpołączeniowym więc aplikacja nie ma jak powiadomić IK-401, aby ta zwolniła pamiętany adres i port źródłowy czyli odblokowała dostęp – dzieje się to automatycznie po odpowiednio długiej (Timeout) „ciszy” na łączu. Dlatego wartość

„Timeout” nie powinna być zbyt duża, najlepiej około 15sek, aby umożliwić dwóm lub większej ilości aplikacji odczytywanie przelicznika w UDP.

Inna aplikacja w UDP „dowiaduje się” o tym, że jej adres IP i port został w IK-401 zapamiętany z faktu otrzymania odpowiedzi z przelicznika. Jeżeli na wysłane zapytanie do przelicznika nie ma odpowiedzi to aplikacja nie wie jaka jest tego przyczyna – może jedynie powtórzyć wysłanie zapytania. Po otrzymaniu odpowiedzi z przelicznika aplikacja nie może także zwlekać z wysłaniem kolejnego zapytania, gdyż po „Timeout” IK-401 „zapomni” adres i port źródłowy bieżącej aplikacji oraz może zapamiętać te dane z innej aplikacji, która jako następna będzie odczytywać przelicznik. Z tego powodu wartość „Timeout” nie powinna być zbyt mała.

Ad pkt 4.

Ten sam przelicznik można odczytać zarówno w TCP jak i w UDP wykorzystując oba kanały szeregowo zarówno w IK-401 jak i w przeliczniku. W IK-401 w jednym z dwóch kanałów wybiera się protokół TCP a w drugim UDP. Jeżeli jednak jeden z kanałów w przeliczniku wykorzystywany jest lokalnie przez Odbiorcę gazu a Dostawca gazu ma dwie asynchronicznie działające aplikacje – jedną, która odczytuje przelicznik w protokole TCP i drugą w UDP to po zawarciu ze sobą obu kanałów RS485 w IK-401 i podłączeniu do kanału 485 w INT-S3 transmisja też będzie działać.

Ad pkt 5.

Żadne przekierowania z WAN do LAN nie są blokowane.

Ad pkt 6. Takie sprawdzenia zostały wykonane w Plum (z dwóch asynchronicznych aplikacji: A i B odczytujących przelicznik)

Nr	Transmisja poprawna	Protokół aplikacji A	Kierunek aplikacji A	Protokół aplikacji B	Kierunek aplikacji B	Komentarz
1	TAK	TCP	z WAN → UART1	TCP	z WAN → UART1	brak
2	TAK	TCP	z WAN → UART2	TCP	z WAN → UART2	brak
3	TAK	TCP	z WAN → UART1	TCP	z WAN → UART2	brak
4	TAK	UDP	z WAN → UART1	UDP	z WAN → UART1	brak
5	TAK	UDP	z WAN → UART2	UDP	z WAN → UART2	brak
6	TAK	UDP	z WAN → UART1	UDP	z WAN → UART2	brak
7	TAK	TCP	z WAN → UART1	TCP	z LAN → UART1	brak
8	TAK	TCP	z WAN → UART2	TCP	z LAN → UART2	brak
9	TAK	TCP	z WAN → UART1	TCP	z LAN → UART2	brak
10	TAK	TCP	z WAN → UART1	UDP	z LAN → UART1	brak
11	TAK	TCP	z WAN → UART2	UDP	z LAN → UART2	brak
12	TAK	TCP	z WAN → UART1	UDP	z LAN → UART2	brak
13	TAK	TCP	z WAN → LAN (Nport 6150) → UART (w NPort)	TCP	z LAN → UART1	brak
14	TAK	TCP	z WAN → LAN (Nport 6150) → UART	UDP	z LAN → UART1	brak
15	TAK	TCP	z WAN → LAN (Nport 6150) → UART	UDP	z LAN → UART2	brak
16	TAK	UDP	z WAN → LAN (Nport 6150, adresacja dynamiczna) → UART	TCP	z WAN → UART1	brak
17	TAK	UDP	z WAN → LAN (Nport 6150, adresacja dynamiczna) → UART	TCP	z WAN → UART2	brak
18	TAK	UDP	z WAN → LAN (Nport 6150, adresacja dynamiczna) → UART	UDP	z WAN → UART1	brak
19	TAK	UDP	z WAN → LAN (Nport 6150, adresacja dynamiczna) → UART	UDP	z WAN → UART2	brak
20	TAK	UDP	z WAN → LAN (Nport 6150, adresacja dynamiczna) → UART	TCP	z LAN → UART1	brak
21	TAK	UDP	z WAN → LAN (Nport 6150, adresacja dynamiczna) → UART	TCP	z LAN → UART2	brak
22	TAK	UDP	z WAN → LAN (Nport 6150, adresacja dynamiczna) → UART	UDP	z LAN → UART1	brak
23	TAK	UDP	z WAN → LAN (Nport 6150, adresacja dynamiczna) → UART	UDP	z LAN → UART2	brak
24	TAK	UDP	z WAN → LAN (Nport 6150, adresacja statyczna) → UART	TCP	z WAN → UART1	brak
25	TAK	UDP	z WAN → LAN (Nport 6150, adresacja statyczna) → UART	TCP	z WAN → UART2	brak
26	TAK	UDP	z WAN → LAN (Nport 6150, adresacja statyczna) → UART	UDP	z WAN → UART1	brak
27	TAK	UDP	z WAN → LAN (Nport 6150, adresacja statyczna) → UART	UDP	z WAN → UART2	brak
28	TAK	UDP	z WAN → LAN (Nport 6150, adresacja statyczna) → UART	TCP	z LAN → UART1	brak
29	TAK	UDP	z WAN → LAN (Nport 6150, adresacja statyczna) → UART	TCP	z LAN → UART2	brak
30	TAK	UDP	z WAN → LAN (Nport 6150, adresacja statyczna) → UART	UDP	z LAN → UART1	brak
31	TAK	UDP	z WAN → LAN (Nport 6150, adresacja statyczna) → UART	UDP	z LAN → UART2	brak