

BECKHOFF

TwinCAT 3 ADS

**Opis protokołu ADS wraz z przykładami komunikacji
pomiędzy dwoma sterownikami**

Wersja dokumentacji 2.0

Aktualizacja: 20.04.2020

Kontakt: *support@beckhoff.pl*

Beckhoff Automation Sp. z o. o.

Spis treści

1.	Komunikacja ADS	4
2.	Konfiguracja sterowników	5
3.	Serwer – program PLC	6
4.	Klient – program PLC	7
5.	Tips & tricks	9
5.1.	Adresowanie	9
5.2.	Nakładanie obszarów pamięci	9
5.3.	Odczytywanie adresu AMS Net Id	11

© Beckhoff Automation Sp. z o.o.

Wszystkie obrazy są chronione prawem autorskim. Wykorzystywanie i przekazywanie osobom trzecim jest niedozwolone.

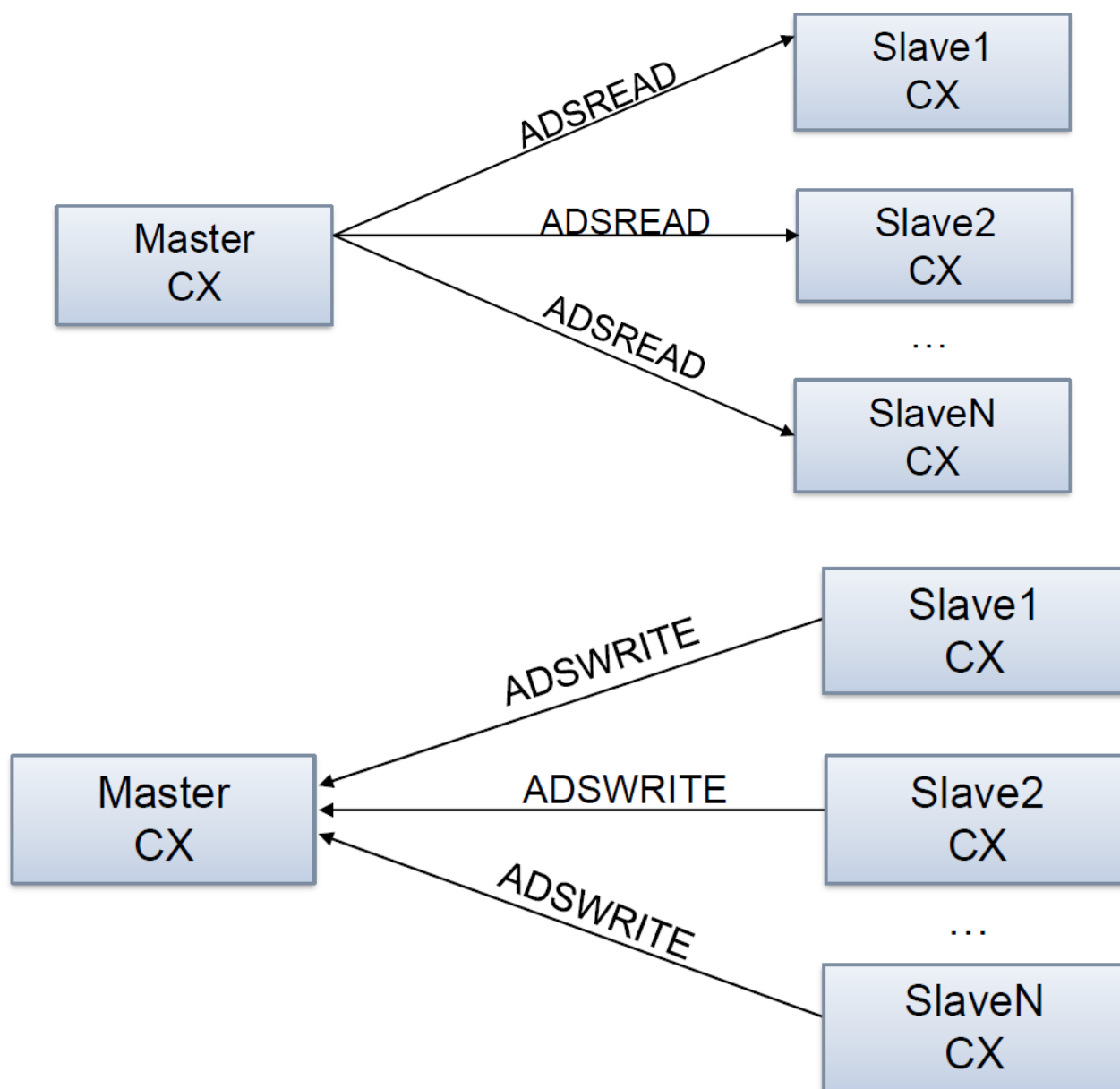
Beckhoff®, TwinCAT®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® i XTS® są zastrzeżonymi znakami towarowymi i licencjonowanymi przez Beckhoff Automation GmbH. Inne oznaczenia użyte w niniejszej prezentacji mogą być znakami towarowymi, których użycie przez osoby trzecie do własnych celów może naruszać prawa właścicieli.

Informacje przedstawione w tej prezentacji zawierają jedynie ogólne opisy lub cechy wydajności, które w przypadku rzeczywistego zastosowania nie zawsze mają zastosowanie zgodnie z opisem lub które mogą ulec zmianie w wyniku dalszego rozwoju produktów. Obowiązek przedstawienia odpowiednich cech istnieje tylko wtedy, gdy zostanie to wyraźnie uzgodnione w warunkach umowy.

Uwaga! Poniższy dokument zawiera przykładowe zastosowanie produktu oraz zbiór zaleceń i dobrych praktyk. Służy on wyłącznie celom szkoleniowym i wymaga szeregu dalszych modyfikacji przed zastosowaniem w rzeczywistej aplikacji. Autor dokumentu nie ponosi żadnej odpowiedzialności za niewłaściwe wykorzystanie produktu. Dany dokument w żadnym stopniu nie zastępuje dokumentacji technicznej dostępnej online na stronie infosys.beckhoff.com.

1. Komunikacja ADS

Protokół ADS to asynchroniczny protokół komunikacyjny wykorzystywany w systemie TwinCAT. Definiuje on komunikację typu klient – serwer. Serwerem nazywamy komputer (sterownik), który udostępnia pewne dane, natomiast klientami nazywamy komputery (sterowniki), które modyfikują dane udostępnione przez serwer. Sieć oparta na protokole ADS jest siecią typu multimaster, co oznacza, że w sieci może występować wiele jednostek nadrzędnych – serwerów. W przypadku sterowników z serii CX możliwe jest nawiązanie nieograniczonej liczby połączeń pomiędzy klientami a serwerami.



2. Konfiguracja sterowników

W celu przesyłania danych pomiędzy dwoma sterownikami za pomocą protokołu ADS, każdy z nich musi mieć dodany drugi sterownik jako route'a. Informacje jak to zrobić można znaleźć w odpowiednim dokumencie na <ftp://ftp.beckhoff.com/poland/Pomoc/Inne> lub poprzez TC3 w zakładce w drzewku projektu System – Routes -> Project Routes -> Add route i dalej jak w [linku](#). Realizacja wymiany zmiennych za pomocą protokołu ADS odbywa się w programie PLC.

Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment
CX_08A38E		10.1.128.114	5.8.163.142.1.1	2.11.2021	Win CE (6.0)	
CX_08B8B2		10.1.128.116	5.8.184.178.1.1	2.11.2021	Win CE (6.0)	
CX_00FE76		10.1.128.18	5.0.254.118.1.1	2.11.2028	Win CE (6.0)	
CX_012228		10.1.128.29	5.1.34.40.1.1	2.11.2035	Win CE (6.0)	
CX_012FC5		10.1.128.144	5.1.47.197.1.1	2.11.2035	Win CE (6.0)	
CX_0215AF		10.1.128.135	10.1.128.135.1.1	3.0.3050	Win XP	
CX_03FC68		10.1.128.112	5.3.252.104.1.1	2.10.1341	Win CE (5.0)	
CX_06AA98		10.1.128.174	10.1.128.174.1.1	3.0.3046	Win XP	

Route Name (Target): CX_0215AF
AmsNetId: 10.1.128.135.1.1
Transport Type: TCP_IP
Address Info: CX_0215AF
☒ Host Name ☐ IP Address
Connection Timeout (s): 1

Route Name (Remote): KNUTG-NB
Target Route: ☐ Project ☒ Static ☐ Temporary
Remote Route: ☐ None ☒ Static ☐ Temporary

Add Route Close

UWAGA! Dla sterowników z TwinCAT 3.1.4024.0 i nowszych dane do logowania to „Administrator” z hasłem „1” dla wszystkich urządzeń (również z Windows CE)!

3. Serwer – program PLC

Możliwy jest odczyt/zapis danych znajdujących się w przestrzeni Memory (zadeklarowanych ... AT %M ...). W pokazanym przykładzie z serwera odczytywana jest zmienna typu STRING zadeklarowana w sposób pokazany poniżej.

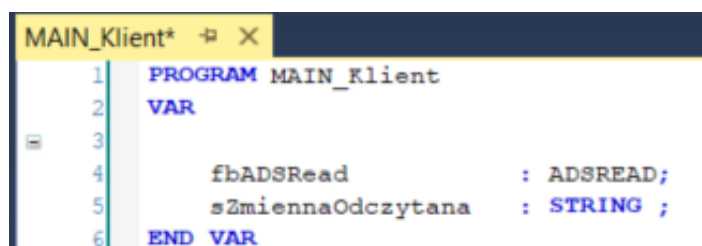
```
MAIN_Serwer  X
1  PROGRAM MAIN_Serwer
2  VAR
3      szmienna    AT%MB0  : STRING := 'Hello world!' ;
4  END_VAR
5
```

4. Klient – program PLC

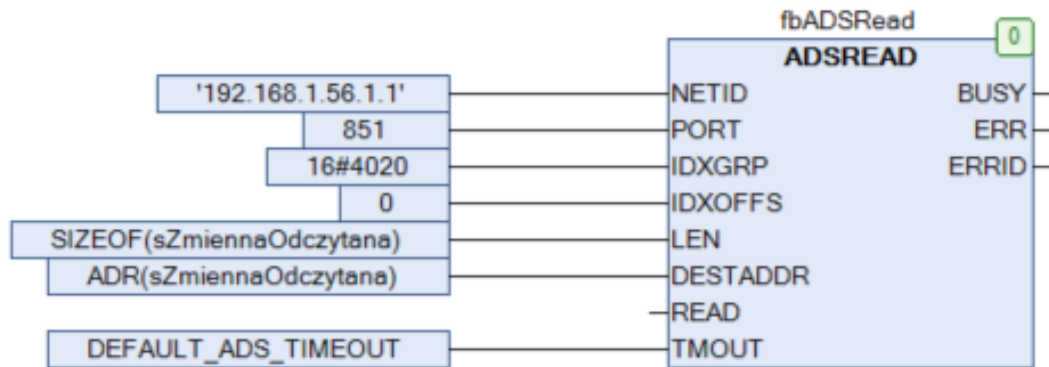
W programie sterownika pełniącego rolę klienta konieczne jest dodanie odpowiedniej biblioteki – Tc2_System.lib.

W programie uruchomianym na kliencie używamy bloku funkcyjnego **ADSREAD**. W bloku tym występują następujące wejścia:

- **NETID (T_AmsNetId)** - Adres ADS sterownika, który jest serwerem,
- **PORT (T_AmsPort)** - Numer portu wykorzystywanego przez PLC Runtime. W przypadku PC i CX dla TC3 dla Runtime 1 jest to port 851, dla Runtime 2 jest to port 852 itd.
- **IDXGRP (UDINT)** - Indeks grupy usługi ADS. Wskazuje on na obszar pamięci z którego dokonujemy odczytu. W przypadku odczytu/zapisu zmiennych przestrzeni Memory (flaga %M) jest to zawsze wartość 16#4020,
- **IDXOFFS (UDINT)** – Offset indeksu usługi ADS. Wskazuje on od którego bajtu chcemy odczytywać pamięć,
- **LEN (UDINT)** - Ilość bajtów które chcemy odczytać. Jest to jednocześnie rozmiar zmiennej do której dokonujemy zapisu (w bajtach) – najczęściej odnosimy się do niej funkcją SIZEOF,
- **DESTADR (DWORD)** - Adres zmiennej do której wpisujemy odczytaną wartość, najczęściej odczytujemy go funkcją ADR,
- **READ (BOOL)** - Wejście reagujące na zbocze narastające, służy do wydawania polecenia odczytu,
- **TMOUT (TIME)** - Limit czasu odczytu, po jego przekroczeniu blok sygnalizuje błąd.
- Wyjścia z bloku są następujące:
- **BUSY (BOOL)** - Wyjście sygnalizujące, że blok wykonuje operację odczytu,
- **ERR (BOOL)** - Wyjście sygnalizujące wystąpienie błędu. Przyjmuje ono wartość TRUE, gdy wyjście BUSY przyjmie wartość FALSE, a odczyt zakończy się niepowodzeniem,
- **ERRID (UDINT)** - ADS Return Code, znaczenie kodu można odnaleźć w dokumentacji protokołu.



```
MAIN_Klient*  X
1  PROGRAM MAIN_Klient
2  VAR
3
4      fbADSRead          : ADSREAD;
5      sZmiennaOdczytana  : STRING ;
6  END_VAR
```



Analogicznie wygląda wywołanie bloku ADSWRITE, z tą różnicą, że jako LEN i SRCADDR podaje się odnośniki do zmiennej, którą chcemy ZAPISAC na wybranym obszarze pamięci w sterowniku - serwerze.

5. Tips & tricks


5.1. Adresowanie

W pokazanym wcześniej przykładzie przesyłana zmienna została zadeklarowana w pamięci jako `AT %MB0`, co oznacza że zostanie ona zapisana w pamięci zaczynając od zerowego bajtu. `AT %MB4` oznaczałoby, że zmienna zostanie zapisana w pamięci zaczynając od piątego bajtu. Inne możliwe deklaracje to:

- `AT %MX` – umożliwia adresowanie pojedynczych bitów, np. `AT %MX4.3` to 4 bit 5 bajtu,
- `AT %MW` – służy do adresowania zmiennych dwubajtowych,
- `AT %MD` – służy do adresowania zmiennych czterobajtowych.

W praktyce zasadne jest używanie jedynie zmiennych deklarowanych jako `AT %MX` oraz `AT %MB`, ponieważ ostatecznie wszystkie zmienne są umieszczane we wspólnym obszarze pamięci więc deklaracje `AT %MB0`, `AT %MW0` i `AT %MD0` dają identyczny rezultat.

Przy deklarowaniu zmiennych należy uważać, aby adres był podzielny przez rozmiar adresowanej zmiennej, np. dla zmiennej typu `REAL` (4-bajtowej) zalecane adresy to `AT %MB0`, `AT %MB4`, `AT %MB8` itd. Podanie niezalecanego adresu skutkuje warningiem, przykład poniżej.

 Variable 'rZmienna' has a granularity of 4 but is located at direct address %MB14 which is not aligned to 4 bytes.

5.2. Nakładanie obszarów pamięci

Przydatną możliwością jest nakładanie jednej zmiennej na kilka innych i odczytywanie tym sposobem kilku zmiennych jednocześnie (za pomocą jednego bloku). Zostało to pokazane na rysunku poniżej. Na tablicę bajtów zostały nałożone różnego typu zmienne. Po stronie klienta i serwera układ zmiennych powinien być taki sam, wówczas rozkodowanie zmiennych odbędzie się samoczynnie.

```
MAIN_Server - X
1  PROGRAM MAIN_Server
2  VAR
3      arrZm      AT%MB0  : ARRAY [1..100] OF BYTE ;
4
5      Zmienna1   AT%MB0  : WORD   := 465      ;
6      Zmienna2   AT%MB2  : INT    := 622      ;
7      Zmienna3   AT%MB4  : REAL   := 423.87    ;
8      Zmienna4   AT%MB8  : WORD   := 321      ;
9      Zmienna5   AT%MB10 : WORD   := 32       ;
10     Zmienna6   AT%MB12 : BYTE   := 120      ;
11     Zmienna7   AT%MB13 : BYTE   := 32       ;
12     Zmienna8   AT%MB14 : WORD   := 892      ;
13 END_VAR
14
```

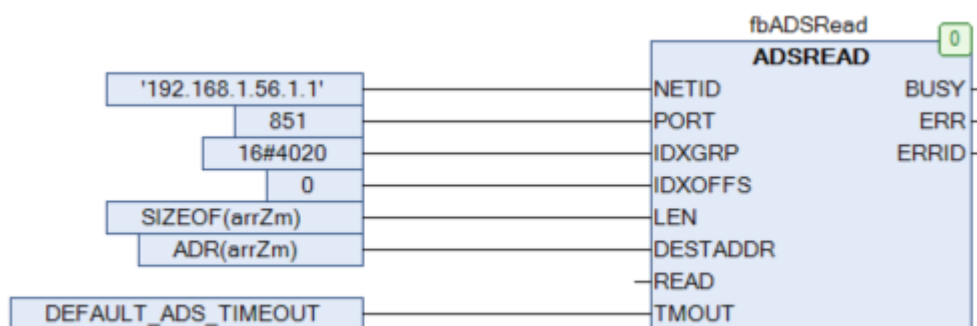
W kliencie do bloku funkcyjnego odczytu/zapisu podpinamy tylko zmienną tablicową – ZM. Pozostałe zmienne uzupełnią swoje wartości samoczynnie. Jest to pokazane poniżej.

Widok offline:

```

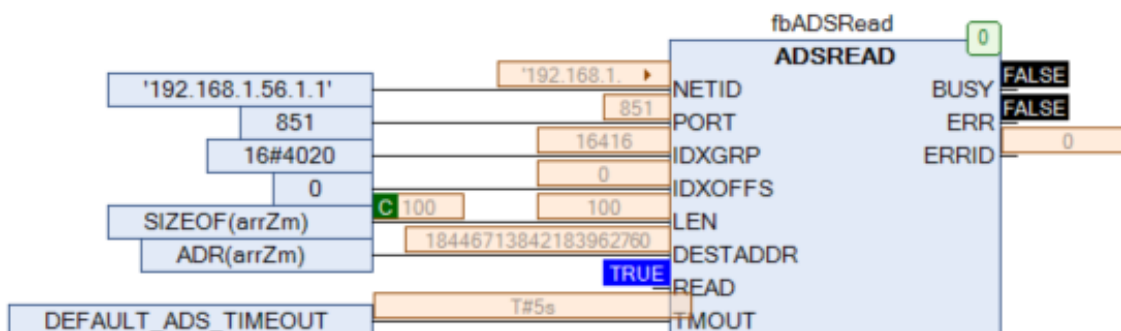
1  PROGRAM MAIN_Klient
2  VAR
3
4      fbADSRead      : ADSREAD;
5      arrZm          AT%MB0 : ARRAY [1..100] OF BYTE ;
6
7      Zmienna1       AT%MB0 : WORD      ;
8      Zmienna2       AT%MB2 : INT       ;
9      Zmienna3       AT%MB4 : REAL      ;
10     Zmienna4       AT%MB8 : WORD      ;
11     Zmienna5       AT%MB10 : WORD     ;
12     Zmienna6       AT%MB12 : BYTE     ;
13     Zmienna7       AT%MB13 : BYTE     ;
14     Zmienna8       AT%MB14 : WORD     ;
15 END_VAR

```



Widok online:

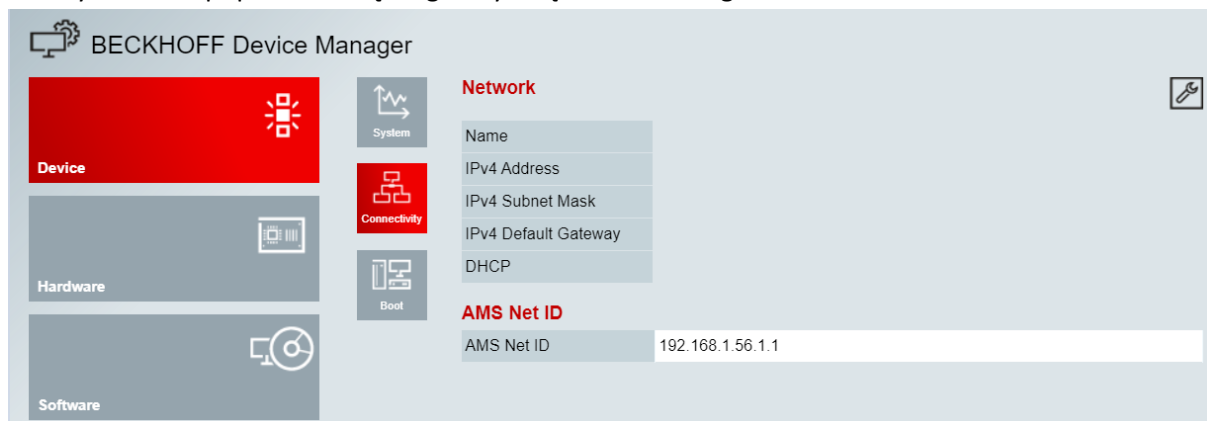
ADS_Instrukcja.Klient_Main.MAIN_Klient				
Expression	Type	Value	Prepared value	Address
fbADSRead	ADSREAD			
arrZm	ARRAY [1..100]...			%MB0
Zmienna1	WORD	465		%MB0
Zmienna2	INT	622		%MB2
Zmienna3	REAL	423.87		%MB4
Zmienna4	WORD	321		%MB8
Zmienna5	WORD	32		%MB10
Zmienna6	BYTE	120		%MB12
Zmienna7	BYTE	32		%MB13
Zmienna8	WORD	892		%MB14



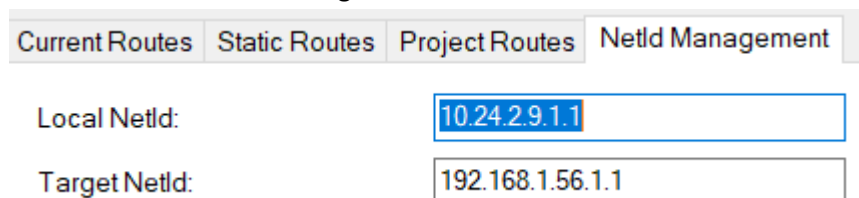
5.3. Odczytywanie adresu AMS Net Id

Adres AMS Net Id sterownika można odczytać w kilku miejscach:

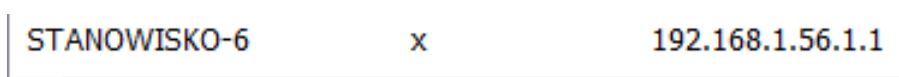
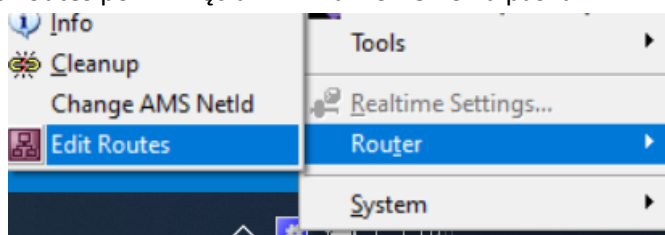
- Jeśli znamy adres IP: poprzez stronę diagnostyczną Device Manager



- W TwinCAT: W zakładce Routes -> NetId Management



- Po wybraniu Router -> Edit Routes po kliknięciu PPM na ikonie TC na pasku:



- W zakładce „Choose Target System”:

