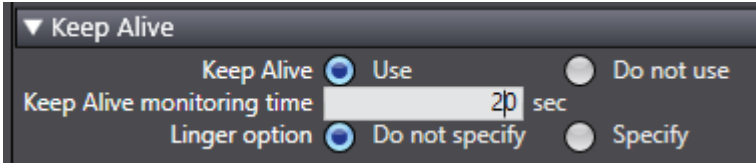


Référence	MTCP_Client_NJ
Révision	2.3
Auteur	JP Viskovic
Date	03/02/2020
+ Support	http://support-omron.fr/

N° Indigo 0 825 825 679
 0.15€ TTC/mm

Modbus TCP Client pour contrôleur NJ

Fonction	Modbus TCP client sur port Ethernet de contrôleur NJ et NX
Connection	<p>The Client_Connect block is used to establish a connection. It has an input Start_Client (normally open contact) and an output Connected (normally open coil). Inputs include IPAddress (value: 192.168.250.55), Port (value: UINT#502), and Cmd_Connect. Outputs include Error_Connect, Err_Code_Client, EtnConnexionProblem, Socket_Client, and Status_Client.</p>
BF Read/Write	<p>This section contains five functional blocks:</p> <ul style="list-style-type: none"> Client_read (MTCP_Client_Fn03): Inputs: Client_Connected (normally open contact), Socket_Client, Unit_ID, Address, Qty, Cmd_ReadReg. Outputs: Read_Ok (normally open coil), Err_Fn03, ErrID_Fn03, Reg. Client_Write1Reg (MTCP_Client_Fn06): Inputs: Client_Connected (normally open contact), Socket_Client, Unit_ID, Address, WORD#16#CAFE, Cmd_Write1Reg. Outputs: Write1Reg_Ok (normally open coil), Err_Fn06, ErrID_Fn06. Client_WriteCoil (MTCP_Client_Fn05): Inputs: Client_Connected (normally open contact), Socket_Client, Unit_ID, Address, CoilValue, Cmd_WriteCoil. Outputs: WriteCoil_Ok (normally open coil), Err_Fn05, ErrID_Fn05. Client_Write (MTCP_Client_Fn10): Inputs: Client_Connected (normally open contact), Socket_Client, Unit_ID, Address, Qty, Registers, Cmd_WriteRegs. Outputs: WriteRegs_Ok (normally open coil), Err_Fn10, ErrID_Fn10. Client_ReadCoils (MTCP_Client_Fn01): Inputs: Connected (normally open contact), Socket_Client, Unit_ID, Address, Qty, Cmd_ReadCoil. Outputs: ReadCoilsOk (normally open coil), Err_Fn01, ErrID_Fn01, Modbus_Coils.
Fichier	MTCP_Client_NJ.zip

Conditions d'utilisation	<p>Le bloc fonction Modbus TCP Client NJ propose certaines fonctionnalités de lecture/écriture conformément aux spécifications définies par l'organisation Modbus.</p> <p>Les blocs fonction MTCP_Client_NJ sont proposés 'tel que' et peuvent servir de base de développement. Les utilisateurs doivent, au préalable, tester leur adéquation avec l'application finale.</p> <p>Omron France ne pourra en aucun cas être tenu pour responsable en cas de dysfonctionnement de l'application finale.</p>																											
Principe	<p>Le FB MTCP_Client_Connect établit une connexion avec le serveur Modbus TCP distant lorsque l'entrée Connect est activée.</p> <p>La sortie Connected autorise l'exécution des blocs de lecture/écriture via l'entrée Enable.</p> <p>Liste des commandes de lecture/écriture :</p> <table border="1" data-bbox="472 689 1402 994"> <thead> <tr> <th>Code</th> <th>Fonction Modbus</th> <th>Bloc Fonction</th> </tr> </thead> <tbody> <tr> <td>0x01</td> <td>Read Coils</td> <td>MTCP_Client_Fn01</td> </tr> <tr> <td>0x02</td> <td>Read Discret Inputs</td> <td>MTCP_Client_Fn02</td> </tr> <tr> <td>0x03</td> <td>Read Holding Registers</td> <td>MTCP_Client_Fn03</td> </tr> <tr> <td>0x04</td> <td>Read Input Registers</td> <td>MTCP_Client_Fn04</td> </tr> <tr> <td>0x05</td> <td>Write Single Coil</td> <td>MTCP_Client_Fn05</td> </tr> <tr> <td>0x06</td> <td>Write Single Register</td> <td>MTCP_Client_Fn06</td> </tr> <tr> <td>0x10</td> <td>Write Multiple Registers</td> <td>MTCP_Client_Fn10</td> </tr> <tr> <td>0x17</td> <td>Read Write Multiple Registers</td> <td>MTCP_Client_Fn17</td> </tr> </tbody> </table> <p>L'option Keep-Alive permet de tester régulièrement le serveur (requête [TCP Keep-Alive])</p> <p>En cas de non-réponse, la connexion est réinitialisée (requête [RST]).</p> <p>Une valeur trop faible peut charger inutilement la communication.</p> 	Code	Fonction Modbus	Bloc Fonction	0x01	Read Coils	MTCP_Client_Fn01	0x02	Read Discret Inputs	MTCP_Client_Fn02	0x03	Read Holding Registers	MTCP_Client_Fn03	0x04	Read Input Registers	MTCP_Client_Fn04	0x05	Write Single Coil	MTCP_Client_Fn05	0x06	Write Single Register	MTCP_Client_Fn06	0x10	Write Multiple Registers	MTCP_Client_Fn10	0x17	Read Write Multiple Registers	MTCP_Client_Fn17
Code	Fonction Modbus	Bloc Fonction																										
0x01	Read Coils	MTCP_Client_Fn01																										
0x02	Read Discret Inputs	MTCP_Client_Fn02																										
0x03	Read Holding Registers	MTCP_Client_Fn03																										
0x04	Read Input Registers	MTCP_Client_Fn04																										
0x05	Write Single Coil	MTCP_Client_Fn05																										
0x06	Write Single Register	MTCP_Client_Fn06																										
0x10	Write Multiple Registers	MTCP_Client_Fn10																										
0x17	Read Write Multiple Registers	MTCP_Client_Fn17																										

1- Variable d'entrée/sortie du bloc MTCP_Client_Connect

Variables d'entrée

Nom	type	plage	Description
Enable	Bool	OFF, ON	Activation du FB
IPAddress	STRING	n.n.n.n	Adresse IP du serveur
Port	UINT	1-65535	Numéro de port du serveur (502 par défaut)
Connect	Bool	OFF, ON	Requête de connexion au serveur

Variables de sortie

Nom	type	plage	Description
Connected	Bool	OFF, ON	ON : Connecté au serveur
Error	Bool	OFF, ON	Drapeau d'erreur
ErrorID	UINT	0 - 65535	Code erreur renvoyé par le socket ou le serveur Modbus TCP (voir liste plus bas)
EtnConnError	Bool	OFF, ON	ON : problème de connexion Ethernet
TCP_Socket	_sSocket	Structure	DstAdr, Handle et SrcAdr

Socket_Status	_eCONNECTION_STATE		Enumerators	Meaning
			_CLOSED	CLOSED status
			_LISTEN	LISTEN status
			_SYN SENT	SYN SENT status
			_SYN RECEIVED	SYN RECEIVED status
			_ESTABLISHED	ESTABLISHED status
			_CLOSE WAIT	CLOSE WAIT status
			_FIN WAIT1	FIN WAIT1 status
			_CLOSING	CLOSING status
			_LAST ACK	LAST ACK status
			_FIN WAIT2	FIN WAIT2 status
	_TIME WAIT	TIME WAIT status		

2- Variables d'entrée des FB

MTCP_Client_Fn01 MTCP_Client_Fn02	type	plage	Description
Enable	Bool	OFF, ON	Activation du FB (utiliser la sortie Connected du FB connect)
TCP_Socket	_sSocket	structure	Instance du socket utilisé (donné par le FB Connect)
Unit_ID	BYTE	00-FF	Unit Identifier (255 par défaut)
Coil_Address	WORD	0 - FFFF	Adresse de la 1 ^{ère} bobine
Coil_Qty	UINT	0 - 00FF	Nombre de bobines
Send_Request	Bool	OFF, ON	Command de lecture

MTCP_Client_Fn03 MTCP_Client_Fn04	type	plage	Description
Enable	Bool	OFF, ON	Activation du FB (utiliser la sortie Connected du FB connect)
TCP_Socket	_sSocket	structure	Instance du socket utilisé (donné par le FB Connect)
Unit_ID	BYTE	00-FF	Unit Identifier (255 par défaut)
Register_Address	WORD	0 - FFFF	Adresse du 1er registre
Register_Qty	UINT	0 - 00FF	Nombre de registres
Send_Request	Bool	OFF, ON	Command de lecture

MTCP_Client_Fn05	type	plage	Description
Enable	Bool	OFF, ON	Activation du FB (utiliser la sortie Connected du FB connect)
TCP_Socket	_sSocket	structure	Instance du socket utilisé (donné par le FB Connect)
Unit_ID	BYTE	00-FF	Unit Identifier (255 par défaut)
Coil_Address	WORD	0 - FFFF	Adresse de la bobine
Set_Value	Bool	OFF, ON	Valeur ON/OFF à écrire
Send_Request	Bool	OFF, ON	Commande d'écriture

MTCP_Client_Fn06	type	plage	Description
Enable	Bool	OFF, ON	Activation du FB (utiliser la sortie Connected du FB connect)
TCP_Socket	_sSocket	structure	Instance du socket utilisé (donné par le FB Connect)
Unit_ID	BYTE	00-FF	Unit Identifier (255 par défaut)
Register_Address	WORD	0 - FFFF	Adresse du registre
Set_Value	WORD	0 - FFFF	Valeur à écrire
Send_Request	Bool	OFF, ON	Commande d'écriture

MTCP_Client_Fn10	type	plage	Description
Enable	Bool	OFF, ON	Activation du FB (utiliser la sortie Connected du FB connect)
TCP_Socket	_sSocket	structure	Instance du socket utilisé (donné par le FB Connect)
Unit_ID	BYTE	00-FF	Unit Identifier (255 par défaut)
Register_Address	WORD	0 - FFFF	Adresse du 1er registre
Register_Qty	UINT	0 - 00FF	Nombre de registre
Registers	ARRAY	0 - FFFF	Source des données (tableau de 128 mots)
Send_Request	BOOL	OFF, ON	Commande d'écriture

MTCP_Client_Fn17	type	plage	Description
Enable	Bool	OFF, ON	Activation du FB (utiliser la sortie Connected du FB connect)
TCP_Socket	_sSocket	structure	Instance du socket utilisé (donné par le FB Connect)
Unit_ID	BYTE	00-FF	Unit Identifier (255 par défaut)
Reg_Addr_Read	WORD	0 - FFFF	Adresse du 1er registre à lire
Reg_Qty_Read	UINT	0 - 00FF	Nombre de registre à lire
Reg_Addr_Write	WORD	0 - FFFF	Adresse du 1er registre à écrire
Reg_Qty_Write	UINT	0 - 00FF	Nombre de registre à écrire
Registers	ARRAY	0 - FFFF	Source des données (tableau de 128 mots)
Send_Request	BOOL	OFF, ON	Commande d'écriture

3- Variables de sortie des FB

Name	type	plage	Description
Cmd_Ok	Bool	OFF, ON	ON : Commande exécutée
Error	Bool	OFF, ON	Drapeau d'exécution
ErrorID	WORD	0 - FFFF	Code erreur renvoyé par le socket ou le serveur Modbus TCP (voir liste plus bas).
Register (Fn03 Fn04 et Fn17 uniquement)	ARRAY		Les valeurs sont renvoyées dans un tableau de 1024 mots ARRAY[0..1023] OF WORD
Coils (Fn01, Fn02, Fn05 et Fn06 uniquement)	ARRAY		L'état des bobines est renvoyés dsans un tableau de 1024 Booléens (ARRAY[0..1023] OF BOOL

Code erreur renvoyé par le serveur Modbus TCP (Modbus exception)

Code		Description
0001	Modbus Exception	ILLEGAL FUNCTION
0002		ILLEGAL DATA ADDRESS
0003		ILLEGAL DATA VALUE
2000	Socket error	Local IP Address Setting Error
2001		TCP/UDP Port Already in Use
2002		Address Resolution Failed
2003		Status Error
2004		Local IP Address Not Set
2006		Socket Timeout
2007		Socket Handle Out of Range
2008		Socket Communications Resource Overflow

Precautions in Using Socket Services

9-7-1 Precautions for UDP and TCP Socket Services

- Communications processing are sometimes delayed when multiple functions of the built-in EtherNet/IP port are used simultaneously or due to the contents of the user program.
- Communications efficiency is sometimes reduced by high communications traffic on the network line.
- The close processing for a close request instruction discards all of the buffered send and receive data for the socket. For example, send data from a send request instruction immediately before the close processing is sometimes not sent.
- After a socket is open, the built-in EtherNet/IP port provides a receive buffer of 9,000 bytes per TCP socket and 9,000 bytes per UDP socket to enable data to be received at any time. If the receive buffer is full, data received by that socket is discarded. Make sure that the user application always executes receive requests to prevent the internal buffer from becoming full.

9-7-2 Precautions for UDP Socket Services

- The destination IP address can be set to a broadcast address for a UDP socket to broadcast data to all nodes on the network. However, in this case, the maximum length of send data is 1,472 bytes. Data lengths broken into multiple fragments (1,473 bytes or more in UDP) cannot be sent.
- For UDP socket, controls to confirm the reliability of communications, such as the confirmation of send data, are not performed. To improve the reliability of communications when you use UDP sockets, make sure the user program confirms that data is sent and resends data when necessary.

9-7-3 Precautions for TCP Socket Services

- If the TCP socket is closed on the remote node without warning during communications (i.e., if the connection is closed), the socket at the local node must also be closed. You can use the Read TCP Socket Status instruction (SktGetTCPstatus) to see if the connection is closed. Immediately close the socket at the local node if the TCP socket at the remote node is closed.
- If the remote node's TCP socket closes without warning, the data to send may remain in the buffer at the local node. The remaining data is discarded in the local node's TCP close processing. The steps that are required in applications to avoid this include sending data from the sending node that permits closing and closing the socket only after checking the remote node.
- While open processing is performed for a TCP socket, a port that was closed first cannot be opened again for 60 seconds from the time the close processing is performed for the remote socket. However, this is not true if you specified 0 (automatic assignment by the Unit) as the port for the SktTCPConnect instruction.
- You can use *Connect* from another socket to open a connection to a socket that was opened with *Accept*. A connection is not opened if you try to use *Connect* from another socket to open a connection to a socket that was opened with *Connect*. Also, a connection is not opened if you attempt to use *Accept* from another socket to open a socket that was opened with *Accept*. Furthermore, you cannot use *Connect* from more than one other node to establish multiple connections with a single TCP socket that was opened with *Accept* on the built-in EtherNet/IP port.
- You can use the keep-alive function for TCP sockets at the built-in EtherNet/IP port. The keep alive function checks whether a connection is normally established when no data is sent or received for a certain period on the communications line where the connection was established. The built-in EtherNet/IP port responds to checks from other nodes even if keep alive is not specified.

Rappel sur le format du protocole Modbus

Lecture de plusieurs bobines Fn01

Exemple : lecture de 19 bits (0001.04 à 0002.06)

Requête		Réponse	
	Donnée		Donnée
Code Fonction	0x01	Code Fonction	0x01
Adresss (poids fort)	0x00	Nbre d'octets	0x03
Adresse (poids faible)	0x14	Bobines 27-20	0xCD
Quantité (poids fort)	0x00	Bobines 35-28	0x6B
Quantité (poids faible)	0x13	Bobines 38-36	0x05

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0CH	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1CH	31 ₁	30 ₀	29 ₁	28 ₁	27 ₁	26 ₁	25 ₀	24 ₀	23 ₁	22 ₁	21 ₀	20 ₁	19	18	17	16
2CH	47	46	45	44	43	42	41	40	39	38 ₁	37 ₀	36 ₁	35 ₀	34 ₁	33 ₁	32 ₀
3CH	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

Les caractères en italique montre la valeur ON/OFF(1/0) du bit.

Lecture de plusieurs registres Fn03

Exemple: lecture de 3 registres (1000 à 1002)

Requête		Réponse	
	Donnée		Donnée
Code Fonction	0x03	Code Fonction	0x03
Adresse (poids fort)	0x03	Nbre d'octets	0x06
Adresse (poids faible)	0xE8	Registre 1000 (poids fort)	0xAB
Quantité (poids fort)	0x00	Registre 1000 (poids faible)	0x12
Quantité (poids faible)	0x03	Registre 1001 (poids fort)	0x56
		Registre 1001 (poids faible)	0x78
		Registre 1002 (poids fort)	0x97
		Registre 1002 (poids faible)	0x13

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1000	A				B				1				2			
1001	5				6				7				8			
1002	9				7				1				3			

Ecriture d'une bobine Fn05

Exemple: écriture d'un bit (0002.02 ON)

Requête		Réponse	
	Donnée	Identique à la requête	Donnée
Code Fonction	0x05	Code Fonction	0x05
Adresss (poids fort)	0x00	Adresss (poids fort)	0x00
Adresse (poids faible)	0x22	Adresse (poids faible)	0x22
Valeur (poids fort)	0xFF	Valeur (poids fort)	0xFF
Valeur (poids faible)	0x00	Valeur (poids faible)	0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0CH	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1CH	31 ₁	30 ₀	29 ₁	28 ₁	27 ₁	26 ₁	25 ₀	24 ₀	23 ₁	22 ₁	21 ₀	20 ₁	19	18	17	16
2CH	47	46	45	44	43	42	41	40	39	38 ₁	37 ₀	36 ₁	35 ₀	34 ₁	33 ₁	32 ₀
3CH	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

Les caractères en italique montre la valeur ON/OFF(1/0) du bit.

Ecriture d'un registre Fn06

Exemple: écriture de &h3AC5 dans le registre 2000.

Requête		Réponse	
	Donnée	Identique à la requête	Donnée
Code Fonction	0x06	Code Fonction	0x06
Adresse (poids fort)	0x07	Adresse (poids fort)	0x07
Adresse (poids faible)	0xD0	Adresse (poids faible)	0xD0
Valeur (poids fort)	0x3A	Valeur (poids fort)	0x3A
Valeur (poids faible)	0xC5	Valeur (poids faible)	0xC5

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2000	3			A			C			5						

Ecriture de plusieurs registres Fn10

Exemple: écriture de 2 des registres 1000 et 1001.

Requête		Réponse	
	Donnée		Donnée
Code Fonction	0x10	Code Fonction	0x10
Adresse (poids fort)	0x03	Adresse (poids fort)	0x03
Adresse (poids faible)	0xE8	Adresse (poids faible)	0xE8
Nbre de registres (poids fort)	0x00	Nbre de registres (poids fort)	0x00
Nbre de registres (poids faible)	0x02	Nbre de registres (poids faible)	0x02
Nombre d'octet de données	0x04		
1ère valeur (poids fort)	0x3A		
1ère valeur (poids faible)	0xC5		
2ème valeur (poids fort)	0x97		
2ème valeur (poids faible)	0x13		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1000	3			A			C			5						
1001	9			7			1			3						

Lecture et écriture de plusieurs registres Fn17

Exemple: lecture de 2 registres depuis l'adresse 1 (1001-2002) et écriture du registre 3 à la valeur CAFE.

Requête		Réponse	
	Donnée		Donnée
Code Fonction	0x17	Code Fonction	0x17
Adresse de lecture (poids fort)	0x00	Nbre d'octets suivants	0x04
Adresse de lecture (poids faible)	0x01	Valeur du registre 1 (poids faible)	0x10
Nbre de registres (poids fort)	0x00	Valeur du registre 1 (poids fort)	0x01
Nbre de registres (poids faible)	0x02	Valeur du registre 2 (poids faible)	0x20
Adresse d'écriture (poids fort)	0x00	Valeur du registre 2 (poids fort)	0x02
Adresse d'écriture (poids faible)	0x03		
Nbre de registres (poids fort)	0x00		
Nbre de registres (poids faible)	0x01		
Nbre d'octets à écrire	0x02		
1ère valeur (poids fort)	0xCA		
1ère valeur (poids faible)	0xFE		

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0000	3			A			C			5						
0001	1			0			0			1						
0002	2			0			0			2						
0003	C			A			F			E						