APPLICATION NOTE

AC500 LINE MODE WITH AC500 V3 CPU’S WORKING WITH A HUGE NUMBER OF MODBUS SOCKETS AT SAME TIME
Contents

1 Introduction ................................................................................................................................................. 3
  1.1 Scope of the document .......................................................................................................................... 3
  1.2 Compatibility ....................................................................................................................................... 3
  1.3 Overview ............................................................................................................................................... 3

2 Using line Mode ........................................................................................................................................ 4
  2.1 Adding Runtime License ...................................................................................................................... 4
  2.2 Configuration in Automation Builder .................................................................................................. 5
  2.3 Setting up Modbus Client FB .............................................................................................................. 5
    2.3.1 FB with Function Codes 3 (Read) and 16 (Write) ......................................................................... 6
    2.3.2 FB with Function Code 23 (Read and Write in one job) ................................................................. 7
  2.4 Using a Step chain for calling the instances ....................................................................................... 8
    2.4.1 Declaration of the Instances ......................................................................................................... 9
    2.4.2 Call of the Instances ....................................................................................................................... 9
1 Introduction

1.1 Scope of the document

Infrastructure Projects like Tunnel need very often a very high number of Modbus Connections at the same time. To fulfill this Requirement AC500 V3 has the opportunity to increase the number of Modbus Sockets with an additional Runtime License “Line Mode”.

1.2 Compatibility

The application note is based on the below engineering system versions. They should also work with newer versions, nevertheless some small adaptations may be necessary, for future versions.

- AC500 V3 PLC FW 3.2.x
- Automation Builder 2.2.1 or newer

1.3 Overview
2 Using line Mode

Without using the Line Mode, the AC500 V3 PLC’s support the following number of Modbus Sockets

<table>
<thead>
<tr>
<th>Model</th>
<th>Sockets</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM5630</td>
<td>30</td>
</tr>
<tr>
<td>PM5650</td>
<td>50</td>
</tr>
<tr>
<td>PM5670</td>
<td>120</td>
</tr>
</tbody>
</table>

Using an additional Runtime License for Line Mode multiplies this Number of Sockets with Factor 5.

2.1 Adding Runtime License

In Automation Builder Project Right Mouse Click to the PLC Node

Choose PLC Runtime Licensing and you will be guided to the Installation/Registration process

This wizard will guide you through the runtime licensing process for:

PLC_AC500_V3
2.2 Configuration in Automation Builder

After successfully activating the License please add "Modbus TCP/IP Client line mode below protocols"

Now the code can be implemented

2.3 Setting up Modbus Client FB

Mostly in such projects there are a lot of Modbus TCP servers with similar functionalities. Therefore it makes sense to create a FB with this functionality and later one you call instances of this FB.
2.3.1 FB with Function Codes 3 (Read) and 16 (Write)

```plaintext
FUNCTION_BLOCK MClient

VAR_INPUT
- MClient_TCP_ADR: DWORD;
- MClient_R_ADDR: WORD;
- MClient_S_ADDR: WORD;
- MClient_NS: WORD;
- MClient_Send_DATA: ARRAY[0..1] OF WORD;

VAR_OUTPUT
- MClient_REC_DATA: ARRAY [0..1] OF WORD;
- MClient_Time: UDINT;
- MClient_Tick_Min: UDINT := 16#FFFFFFF;
- MClient_Tick_Max: UDINT := 0;
- MClient_R_OK: WORD;
- MClient_R_NOK: WORD;
- MClient_Last_Errno_R: AC500_ModbusTCP.ERROR_ID;
- MClient_S_OK: WORD;
- MClient_S_NOK: WORD;
- MClient_Last_Errno_S: AC500_ModbusTCP.ERROR_ID;

VAR
- Step: INT;
- M1: ModbusMaster2;
- Start_Time: UDINT;
- End_Time: UDINT;
- Read_data: ARRAY [0..99] OF WORD;
- Send_data: ARRAY [0..99] OF WORD;
- MClient_S: WORD;

VAR_INPUT
// call Modbus master with current inputs

IF Step=0 AND M1.Burst=true THEN \\
  // abort read data from Modbus TCP server
  M1(Reaction:=TRUE, Rebus:=TRUE, MClient_TCP_ADR, 0,0,0,0,0,0,MClient_R_ADDR, Shoc=MClient_NS, Data:=READ(Read_data), KeepTimeout:=0000, Deadline:=100, hypervisor:=AC500_ModbusTCP.ERROR_ID, VERIFY:=0, Rebus=M1.Value:=TRUE); // Read current time in my system
  Step:=10; // go to wait read ready
ENDIF;

IF Step=0 AND M1.Burst=true THEN \\
  // wait for ready read data -> SUCCESS
  M1[Read_data[1]:=Read_data[2];]
  M1[Read_data[2]:=Read_data[3];]
  M1[ModbusTCP.ERROR_ID]:=M1[Read_data[2];]
  Step:=10; // go to "write data"
ENDIF;

IF Step=0 AND M1.Burst=false THEN \\
  // wait for ready read data -> fail
  M1[Read_data[1]:=M1[Read_data[2];]
  M1[Read_data[2]:=M1[Read_data[2];]
  Step:=10; // go to start new cycle - "read data"
ENDIF;
```

6 3ADRO10434,1, en_US
AC500 LINE MODE WITH AC500 V3 CPU’S

2.3.2 FB with Function Code 23 (Read and Write in one job)

```plaintext
FUNCTION_BLOCK Mod_Client_FCT23

VAR_INPUT
MClient_TCP_Addr: STRING;
MClient_R_ADDR: WORD;
MClient_S_ADDR: WORD;
MClient_NB: WORD;
MClient_Send_DATA: ARRAY [0..1] OF WORD;

END_VAR

VAR_OUTPUT
MClient_Rec_DATA: ARRAY [0..1] OF WORD;
MClient_Time: UDINT;
MClient_Time_Min: UDINT := 16#FFFFFFF;
MClient_Time_Max: UDINT := 0;
MClient_OK: WORD;
MClient_BOG: WORD;
MClient_Last_Error: AC500_ModbusTCP.ERROR_ID;

END_VAR

VAR
MClient_ModBus_TCP;
Funct_23: ETH_MOD_FCT23_TYPE;
Start_Time: UDINT;
step: INT;
End_Time: UDINT;

END_VAR
```

3ADR010434, 1, en_US 7
2.4 Using a Step chain for calling the instances

Working with a high number of Modbus Client instances (e.g. 240) create a high CPU Load if calling all the instances to the same time. To avoid this high load, we recommend calling the FB’s in a step chain. That mean call the first 50 instances, wait a dedicated time (e.g. 20ms) and then call the next number of Instances.
Declaration global or local in IEC program:
Instance (=socket) of POU ModTcpMast2 for each server, e.g.:  
fbModMast_11, fbModMast_12, ..., fbModMast_1X  
fbModMast_21, fbModMast_22, ..., fbModMast_2X

..  
fbModMast_1Y, fbModMast_2Y, ..., fbModMast_YX

Note: set input KeepAlive >= runtime for all Steps 1..Y  
=> socket should not be closed during runtime of all steps

Call in IEC program – step chain with „Y“ steps:
Step1: (* call all POU\s of cycle 1 *)  
  fbModMast_11(Execute := TRUE);  
  fbModMast_12(Execute := TRUE);  
  fbModMast_13(Execute := TRUE);

..  
  fbModMast_1X(Execute := TRUE);
Step2: (* call all POU\s of cycle 2 *)  
  fbModMast_21(Execute := TRUE);  
  fbModMast_22(Execute := TRUE);  
  fbModMast_23(Execute := TRUE);

..  
  fbModMast_2X(Execute := TRUE);
StepY: (* call all POU\s of cycle Y *)

2.4.1 Declaration of the Instances

2.4.2 Call of the Instances

Start with the first instances
After calling the first instances wait for 20ms and call the next

Go back to Start