CP500 CONTROL PANELS

CP500 Soft: Recipe
(Example with CP503 & AC500)
Step7
Summary

1. Material used .................................................................................................................. 3
2. New project and settings ................................................................................................. 4
3. Create “Recipe block” ....................................................................................................... 5
4. Load, save and delete recipe ........................................................................................... 6
5. Show the result .................................................................................................................. 6
6. AC500 Modbus variable range ......................................................................................... 7
7. Help: Modbus configuration with PS501 ....................................................................... 8
1. **Material used**

Material used:

- CP503  (Reference: 1SBP260172R1001)
- CAB5  (Reference: 1SBN260210R1001) or CAB6 (Reference: 1SBN260211R1001) on RS422 port
- CAB57  (Reference: 1SBN260215R1001)
- CP500Soft  (Reference: 1SBS260283R1001)
- PM581-ETH  (Reference: 1SAP140100R0170)  CPU
- TB521-ETH  (Reference: 1SAP112100R0170)  Terminal Base
- DC532  (Reference: 1SAP240100R0001)  16DI / 16DO
- 24VDC Power supply
2. **New project and settings**

For a first use with AC500 please refer to Step6 : First Step with AC500.

1- Open the CP500 Programmer

2- Create a new project with CP503 and “AC31 40/50 series Serial Driver”.

3- Go on “Setup / Recipe settings”

In “Recipe Ctrl Block:”, use the Address of the register that contains information from the recipe.
Our example: %MW0.0

In “Current Recipe Reg:”, use the address of a second register, which contains the Name of the active recipe.
Example: %MW0.1

4- Create a second page named “Recipe” with a jump or push button to go on it.
3. **Create “Recipe block”**

5- Open the “Recipe” block

6- Put on the page the information beside:

Addresses of the variables used:

Dr1 with the Address %MW0.1000 (0.3 object)
Dr2 with the Address %MW0.1001 (0.3 object)

Output0 with the Address %MX0.1020.0 (0/1 object)
Output1 with the Address %MX0.1020.1 (0/1 object)

For this don’t forget to cross “Enable Operator input” in “Access”

To visualise the outputs on the AC500, %MX0.1020.0 and %MX0.1020.1 must be affected to the outputs available on the DC532 extension, example:

%MX0.1020.0 = %QX0.0
%MX0.1020.1 = %QX0.1

Object (It’s ASCII text). It will be the Name of the recipe loaded.

Address %MW0.1

All the variables in this page will be saved or loaded in the recipe when clicking on the button F2, F3 or F4 (See next page).
4. **Load, save and delete recipe**

7- Button for recipe functionalities:

Double click on F2.
It’s opening the window beside.

Select F2.
Choose in the list of “Other function”: “Load recipe”
Update.

Select F3.
Choose in the list of “Other function”: “Save recipe”
Update.

Select F4.
Choose in the list of “Other function”: “Delete recipe”
Update. And OK.

5. **Show the result**

8 – Verify in “Setup / Peripherals” that the address and the communication parameters from the PLC are right.

9 - Send the program in the CP503

10 - Change the value go with the arrow on the value you want to change, enter change it and enter again.

11 – When you have entered all your variables, press F3 to save your recipe. Enter the name you want for it. Make a second recipe in doing the same. After, you can load all recipes you have saved with F3. The name of the recipe you load appear near Recipe Name.
6. **AC500 Modbus variable range**

Ranges access for the different CPUs

**PM571: 4 kB**
1 segment of variables available:
- %MX0.0 to MX0.4095.7
- %MB0.0 to %MB0.4095
- %MW0.0 to %MW0.2047
- %MD0.0 to %MD0.1023

**PM581 and PM591: 128 kB**
2 segments of variables available:
- %MX0.0 to MX0.65535.7
- %MB0.0 to %MB0.65535
- %MW0.0 to %MW0.32767
- %MD0.0 to %MD0.16383
- %MX1.0 to MX1.65535.7
- %MB1.0 to %MB1.65535
- %MW1.0 to %MW1.32767
- %MD1.0 to %MD1.16383

**WARNING**
If you use data table the data could not begin in segment “1” and finish in segment 2. It will create an error message.

**Structure extraction**

<table>
<thead>
<tr>
<th>BYTE</th>
<th>Bit (byteorientiert)</th>
<th>Wort (WORD)</th>
<th>Doppelwort (DWORD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>%MB0.0</td>
<td>%MX0.0.0 .. %MX0.0.7</td>
<td>%MW0.0</td>
<td>%MD0.0</td>
</tr>
<tr>
<td>%MB0.1</td>
<td>%MX0.1.0 .. %MX0.1.7</td>
<td>%MW0.1</td>
<td>%MD0.1</td>
</tr>
<tr>
<td>%MB0.2</td>
<td>%MX0.2.0 .. %MX0.2.7</td>
<td>%MW0.2</td>
<td>%MD0.2</td>
</tr>
<tr>
<td>%MB0.3</td>
<td>%MX0.3.0 .. %MX0.3.7</td>
<td>%MW0.3</td>
<td>%MD0.3</td>
</tr>
<tr>
<td>%MB0.4</td>
<td>%MX0.4.0 .. %MX0.4.7</td>
<td>%MW0.4</td>
<td>%MD0.4</td>
</tr>
<tr>
<td>%MB0.5</td>
<td>%MX0.5.0 .. %MX0.5.7</td>
<td>%MW0.5</td>
<td>%MD0.5</td>
</tr>
<tr>
<td>%MB0.6</td>
<td>%MX0.6.0 .. %MX0.6.7</td>
<td>%MW0.6</td>
<td>%MD0.6</td>
</tr>
<tr>
<td>%MB0.7</td>
<td>%MX0.7.0 .. %MX0.7.7</td>
<td>%MW0.7</td>
<td>%MD0.7</td>
</tr>
<tr>
<td>..</td>
<td>..</td>
<td>..</td>
<td>..</td>
</tr>
<tr>
<td>%MB0.65532</td>
<td>%MX0.65532.0 .. %MX0.65532.7</td>
<td>%MW0.32767</td>
<td>%MD0.16383</td>
</tr>
<tr>
<td>%MB0.65533</td>
<td>%MX0.65533.0 .. %MX0.65533.7</td>
<td>%MW0.32767</td>
<td>%MD0.16383</td>
</tr>
<tr>
<td>%MB0.65534</td>
<td>%MX0.65534.0 .. %MX0.65534.7</td>
<td>%MW0.32767</td>
<td>%MD0.16383</td>
</tr>
<tr>
<td>%MB0.65535</td>
<td>%MX0.65535.0 .. %MX0.65535.7</td>
<td>%MW0.32767</td>
<td>%MD0.16383</td>
</tr>
</tbody>
</table>
7. **Help: Modbus configuration with PS501**

**COM1 and COM2 must define in menu “Ressources / PLC configuration / Interfaces”**

*Select Modbus configuration for COM1 and COM2*

**COM1 and COM2 are configured in Modbus (RS232 or RS485)**

**Configuration of COM parameters**

**Warning:** - Necessary to select “telegram” for RTS control parameter if RS485 is used. - Always used 8 data bits selection