



HVAC&CR Community meeting

New Features in AVP – 2018 April 18th

Presented by: Pierpaolo Armeli
HVAC Solution Center



New Features in AVP

- 3 AVP ModBUS Master Features
- 9 AVP HMI Global Variables Trick
- 10 AVP as ModBUS Master Sample

AVP ModBUS Parameters

New ModBUS parameter for the AVP

The AVP Touch Display has the following ModBUS Parameters available from Device:

RS485 On Board

Address	Name	Value	Um	Default	Min	Max	Description
8198	Addr_RS485_OB	1	num	1	0	247	RS485 On Board address
8199	Proto_RS485_OB	1=Modbus/RTU master	num	1=Modbus/RTU master			Select RS485 On Board protocol
8200	DataBit_RS485_OB	8	num	8	8	8	RS485 On Board Data bit number
8201	StopBit_RS485_OB	1	num	1	1	2	RS485 On Board stop bit number
8202	Parity_RS485_OB	2=Even	num	2=Even	0	2	RS485 On Board parity protocol
8203	Baud_RS485_OB	2=38400	num	2=38400	0	4	RS485 On Board baud rate protocol

ModBUS Parameters of the M172DCL on SoMachine Device

The new **Proto_RS485_OB** parameter allows to set the RS485 device port as either ModBUS Master or ModBUS Slave.

This parameter can be changed run time.



ModBUS Commands on Event: Application

New ModBUS Master Functions in Application for AVP

- The AVP with target 659.4 will have new ModBUS Master target blocks available in **Application**.

These functions allow to send **ModBUS commands on Event** through the master RS485 port by calling them in Application.

The messages can be send **only** from programs assigned to the **Background** task.

- The message that has to be sent will be read from the new function by the given *object pointer*.

The pointer has a @INT data type, therefore **shorter/longer messages** (e.g. USINT, UDINT) to be sent **have to be first stored/split in an INT variable**.

Then, the pointer to this last one has to be passed to the function.

Operators and blocks		
Name	Type	Description
 sysMbMRTU_FC01	Function	Send 0x01 command.
 sysMbMRTU_FC02	Function	Send 0x02 command.
 sysMbMRTU_FC03	Function	Send 0x03 command.
 sysMbMRTU_FC04	Function	Send 0x04 command.
 sysMbMRTU_FC05	Function	Send 0x05 command.
 sysMbMRTU_FC06	Function	Send 0x06 command.
 sysMbMRTU_FC15	Function	Send 0x0F command.
 sysMbMRTU_FC16	Function	Send 0x10 command.

Function: sysMbMRTU_FC06 (ver. 1.0.0. EMBE...)

Return Value: UINT

Input:

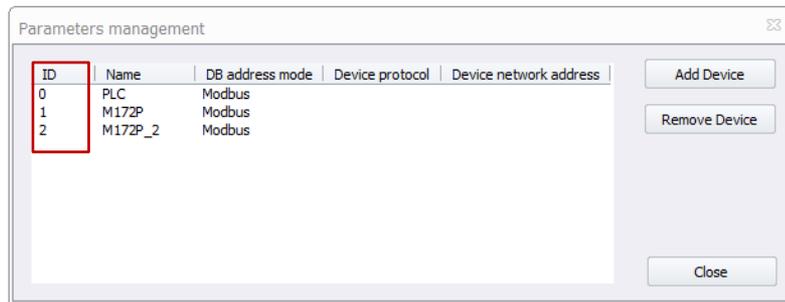
Name	Type	Description
addr	USINT	Physical address of the target slave
base	UINT	Address of the Register to write
object	@INT	Register pointer
timeout	UINT	Timeout [ms]

The Basic_Addons Library will allow easier data conversion

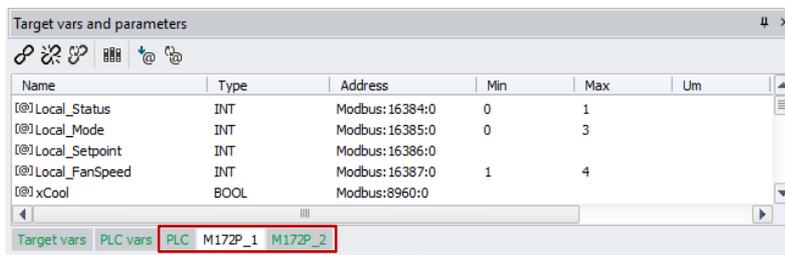
ModBUS Commands: User Interface

New ModBUS Master Functions in User Interface for AVP

- The AVP allows to link up to eight (8) Slaves ModBUS variables in User Interface, by using their related *.parx* file.
- Each Slave is represented by an ID:
 - The ID number can be edited from the *Parameters management* window in User Interface.
 - **Only IDs ranging from 1 to 8 are supported.**
 - The **ID number 0** is always associated to the *parx* of the **AVP itself.**
- The ModBUS Address of each of the Slaves (1-8) has to be specified through a parameter (in Device), which cannot be changed in real time (a reboot is required).



Parameters management in User Interface



Sample of target vars and parameters of one Slave (using *parx*)

Remote Slave 1							
Address	Name	Value	Um	Default	Min	Max	Description
8254	RemoteSlave1_addr	1	num	1	0	247	Remote Slave 1 address
8262	RemoteSlave1_32bit_order	0=Word NOT s	num	0=Word NOT swapped	0	3	Remote Slave 1 32bit order

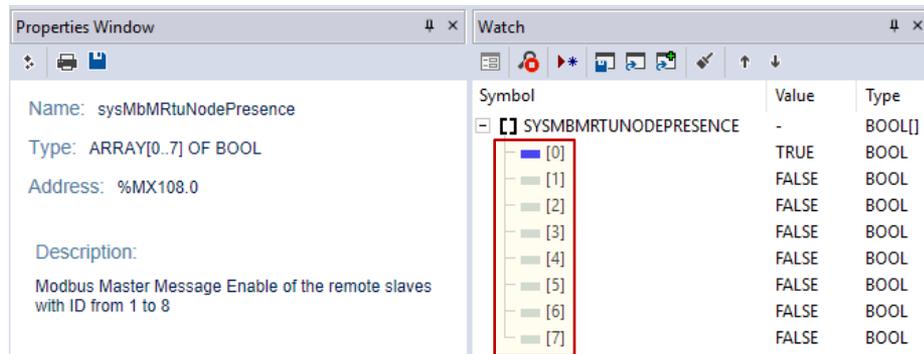
ModBUS Commands: User Interface vs Slave Node Presence

New ModBUS Master Functions in User Interface vs Slave Node Presence

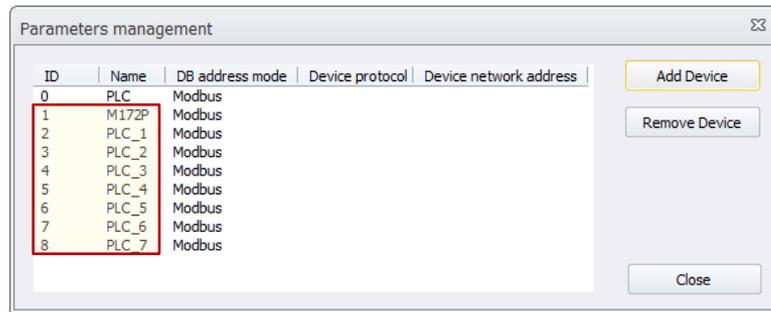
Regarding the **ModBUS commands executed from User Interface** to the up to eight Slaves, it is possible to manage the **presence** status of each Slave, in order to avoid communication delays.

The presence of the nodes whose IDs range from 1 to 8 is defined by the **sysMbMRtuNodePresence** array and it has to be changed manually by the developer **in Application**.

Even tough this edit has to be done in Application, it is important to clarify that it has only effect on the ModBUS commands that take place in User Interface.



The node numbers defined in User Interface are scaled by 1 with respect to the sysMbMRtuNodePresence array row number.



ModBUS Commands: Application vs User Interface

New ModBUS Master Functions in Application vs User Interface for AVP

APPLICATION

The new ModBUS on Event functionality of the AVP, which is available in Application, allows the developer to **define exactly when the ModBUS command has to be sent** (on Event) and supports **all the FC family** commands.

The ModBUS commands sent from Application will be executed in the Background task and therefore **will block the Background task** during their execution.

USER INTERFACE

On the other side, in User Interface **only Holding Registers** reads/writes are supported and **only single register read/write** operations are executed.

Therefore, potential optimization using multiple registers reads/writes in Application has to be taken into account before choosing to use User Interface for a each specific ModBUS communication.



ModBUS Commands on Event: Application

New ModBUS Master Functions in Application vs Background Task

It is also important to **evaluate the command *timeout*** used when calling the new functions, as it will block the Background task for the whole *timeout* period in case of any communication delay/error.

```
sysMbMRTU_FC16(0,ADR_sysClockSet_seconds,8,ADR(i_sysClock),1000);
```

In addition, it has to be checked if any **slave is unreachable**: the **commands** sent to the unreachable slaves should be **skipped**, otherwise the Background task will be blocked for the whole *timeout* period of each command.

It is a good practice **not to send too many ModBUS messages** per each background task cycle execution.



AVP HMI Global Variables Trick

New AVP Features: Global Variables in User Interface accessible directly from Application

When a project with the AVP target is compiled, a merge of the PLC and the HMI projects takes place. As a consequence, the Global Variables defined in User Interface are directly accessible from Application, by using the variables name. However, they are not visible/listed in Application.

In this way, **in order to have the same variable available both in Application and in User Interface**, on the AVP target **it is not anymore required to** declare the variable in Application and to:

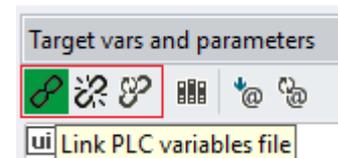
- use the @ syntax when calling the variable in an object property;
- use the *Video.GetParam* and *Video.SetParam* functions in User Interface to read/write on it from a script, as scripts do not support the @ syntax.

It is actually enough to declare the variable in User Interface as Global Variable and then the variable will be accessible in Application and in User Interface just by calling its name, without any additional syntax required.

The only prerequisite is to link the PLC .exp variables file in User Interface.



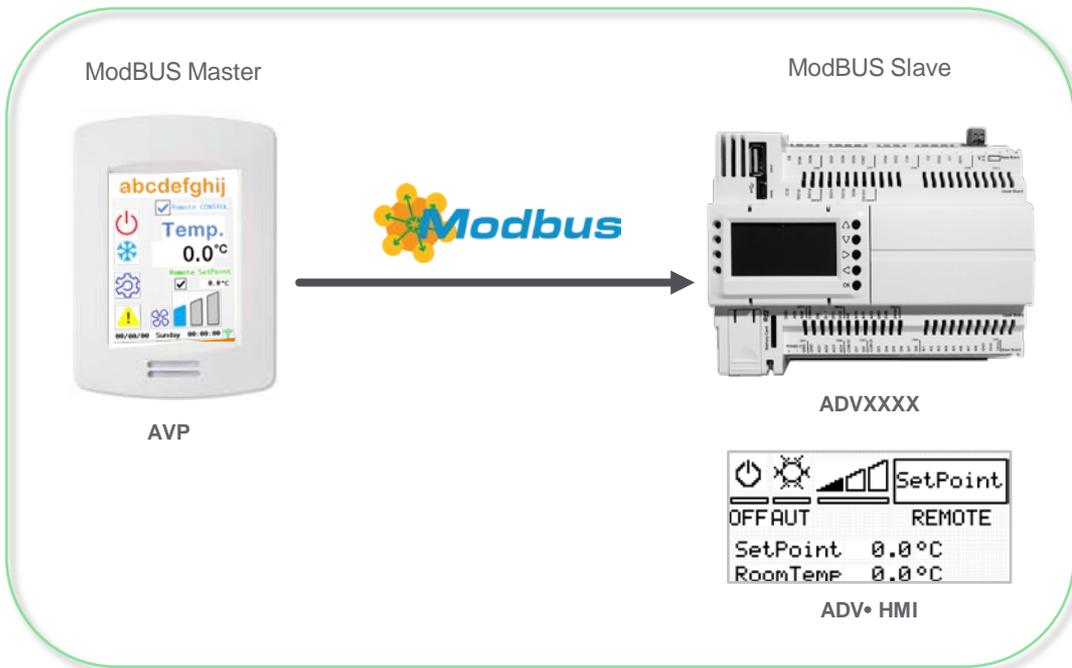
.exp



Sample: AVP (Master) is connected to a Fan Coil (Slave)

Fan Coil using Advance (Slave) is monitored/controlled by an AVP (Master)

The sample shows how an **AVP** set as **ModBUS Master** can be used to monitor and/or control a **Fan Coil (ModBUS Slave)**, which in this case is represented by an AVDXXXX (Advance).



```
SendingTrigger(ck:= DelaySending.Q);
EnableWrite:= SendingTrigger.q;

if (EnableWrite) then
  MBusWrites_FC16.object1:= Remote_Status_HMI;
  MBusWrites_FC16.object2:= Remote_HC;
  MBusWrites_FC16.object3:= TO_UINT(Remote_Setpoint);
  MBusWrites_FC16.object4:= Remote_FanSpeed;
  MBusWrites_FC16.object5:= Remote_type;
  MBusWrites_FC16.object6:= Remote_Active;

  MBusWrites_FC16(addr:= Slave_Addr, base:= 8969 + 0
no:= 6, timeout:= 1000, waitBeforeS
end_if;
```

AVP
Application code sample

```
if Remote_Active then
  ChangeStatus:= TRUE;
  case Remote_FanSpeed of
4: Remote_FanSpeed:= 1;
1: Remote_FanSpeed:= 2;
2: Remote_FanSpeed:= 3;
3: Remote_FanSpeed:= 4;
end_case;
else
  Remote_FanSpeed:= 4;
end_if;
```

AVD
User Interface
code sample



Thank you for your attention.

The chat is ON , if you have -more- questions.





Life Is On