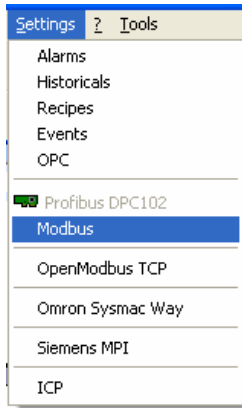


**Serial MODBUS
On several comports
Visual IO**

Ver 7.xx
ARSoft international - Copyright © 1999

Serial Modbus configuration



Principle:

This grid of parameters describes the recurrent communications. Those communications can read or write bits and words (16 bits) in Modbus slaves. Before describing the communications, you must declare a list of Words (WORD) and bits (BOOLEAN) receiving or reflecting the states variables in the slave equipment. After validation by clicking on the button OK, the system generates the final Modbus compiled driver automatically.



Note: A right click with the mouse allows copying and duplicating lines.

Example:

I want to read 10 words in a slave equipment number 2.

1/ Declaration of 10 Words in global variables.

```
W1,W2,W3,W4,W5,W6,W7,W8,W9,W10 : Word ; Or
Tablow : Array [1..10] Of Word ;
```

The first line allows reading in the slave #2 of 10 consecutive words (16 bits) starting from the address 200 in Hexa. The values of these words will be stored in W1 to W10. The column Variables defines the first word for the reception or the emission. If several words are read, the first word will be stored into the specified variable then the following into the words consecutively declared in the global variables.

The column Command (optional) set a bit which will be reset by Modbus communication. The Bit must be set (True) to activate the concerned communication (Read/Write Bits or Words). If the dialog is ok, the system resets this bit (false = receipt acknowledgement).

Specifications

Generally when your application reads Words (16Bits), High/low bytes are inverted at the reception because the slaves return words in Motorola Format (in Modbus bible). Windows and Visual IO use Intel format. If in special case you need to disable this function, by program write

Set Intel Format by program (exceptional).

In a form write

```
Uses Modbus ;
ModbusSerialFormat := True ;
```

Set Modbus Time out

The time out in reception is set at 800ms; if the slave doesn't reply in 800ms the transaction is cancel. You can modify this value by program.

```
Uses Modbus ;
MODBUSRTO := 1000; /** Here the time out is set at 1sec (1000 Msec)
```

Reading Writing Floating points.

You can read or write floating points in modbus slaves.

You must define Single variables in visual IO

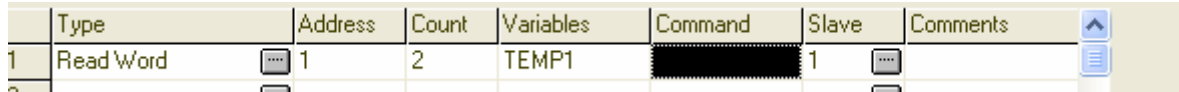
Example:

Temp1 : **Single**;

Temp2 : **Single**;

Etc..

In the grid of parameters you must specify the variable (Temp1) and count = 2 for each var.



	Type	Address	Count	Variables	Command	Slave	Comments
1	Read Word	1	2	TEMP1		1	

Because a floating point is formatted on 4 bytes (= 2words).

The MODBUS Driver

After validation (button Ok), Visual I/O generates a compiled driver regarding of this configuration. This driver is implemented automatically in the current application (Menu: constitution application / Priority tasks). This driver will be executed in the priority task to perform a communication at maximum Speed.

The program name generated is **PCMODBUS.VPU** (source PCMODBUS.PAS is readable).

Cancelling the driver

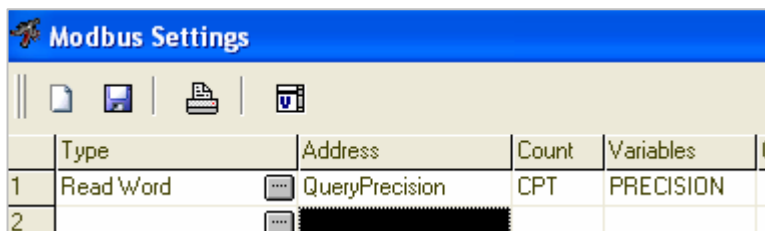
Go to the menu utilities, Constitution Application and priority task tab. Double click on **PCMODBUS** to remove it from the list.

Notations

If you need to write a value in Hexadecimal you must write \$ before the number.

Example : \$200 or \$1A (200 & 1A in hexa)

You can also determine a variable in the column address or Count and not a constant.



	Type	Address	Count	Variables
1	Read Word	QueryPrecision	CPT	PRECISION
2				

Here the *Query* global variable will give the final address to read in the Modbus slave. You'll have to affect the Query variable by program. You can also initialise the global variable with starting value:

Example : Query : Integer=100;

It is the same principle for Count column with the CPT variable indicate the number of word to read in the slave.

Modbus Codes

Modbus use function codes in its protocol.

01 Read Coil Status

02 Read Input Status

03 Read Holding Registers = READ word in visual IO

04 Read Input Registers = READ Internal Word in visual IO

Some slave specify the Modbus code in theirs address example address = 300018. You must understand Code 3 and Adress 018. Address = 412123. You must understand Code 4 and Address 12123.

If you specify address 400018 in the column address with Read Word, Visual IO extract the 6th number (here 4) and affect this number to the Modbus code. In this case you can read with code 01 and 02.

In visual IO the notation will be **2xxxxx**.

How to create to independent drivers

It seems necessary for certain cases to have a communication on COM1 and another communication on COM2 and so on.

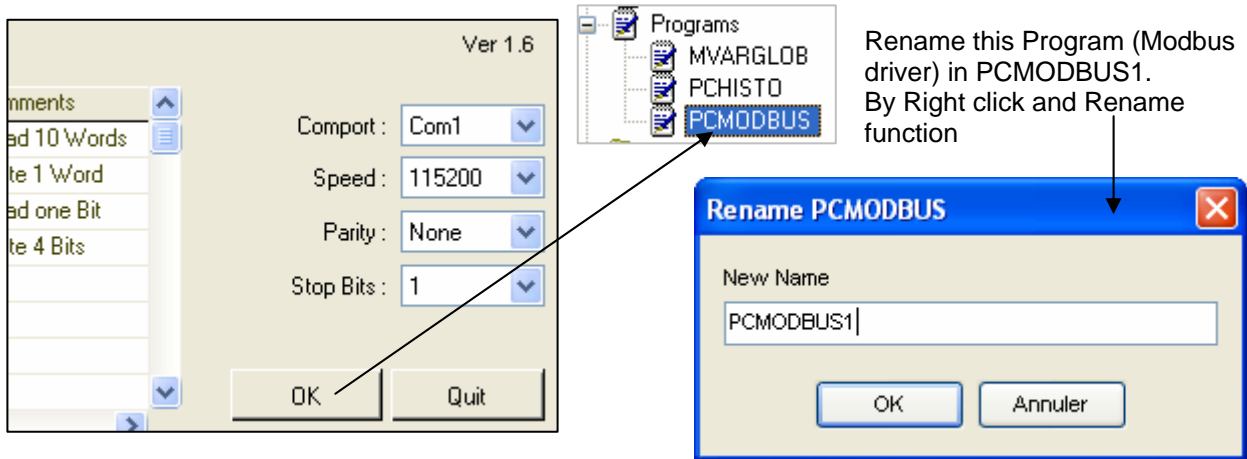
The Modbus setup allow only one communication because the grid is not planned to set different comport at the same time.

To have two (or more) different communications driver you must rename the **PCModbus.*** in **PCMODBUS1.*** and make again a new configuration. You can create PCmodbus1, PCmodbus2 .. PCModbusX.

Operating mode for 2 different comports

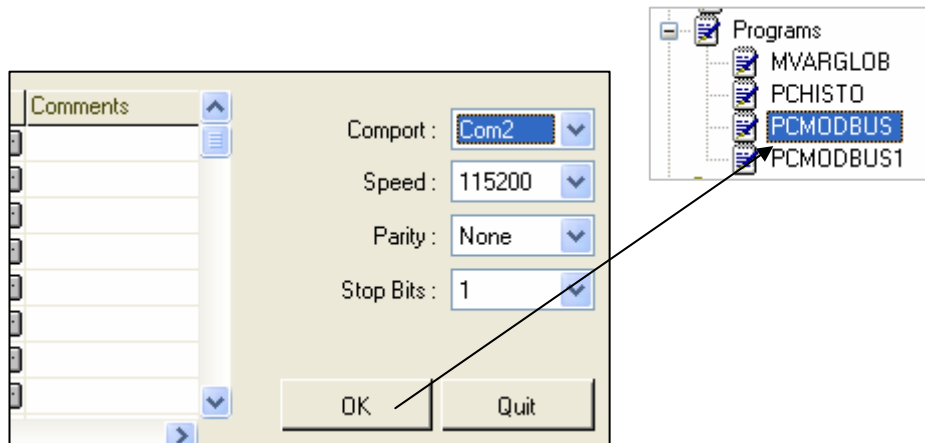
1/ Create your first settings (on com1)

2/ Click Ok Button – The system generates PCMODBUS.VPU (in the vio navigator below)



Visual IO renames PCMODBUS in PCMODBUS1.

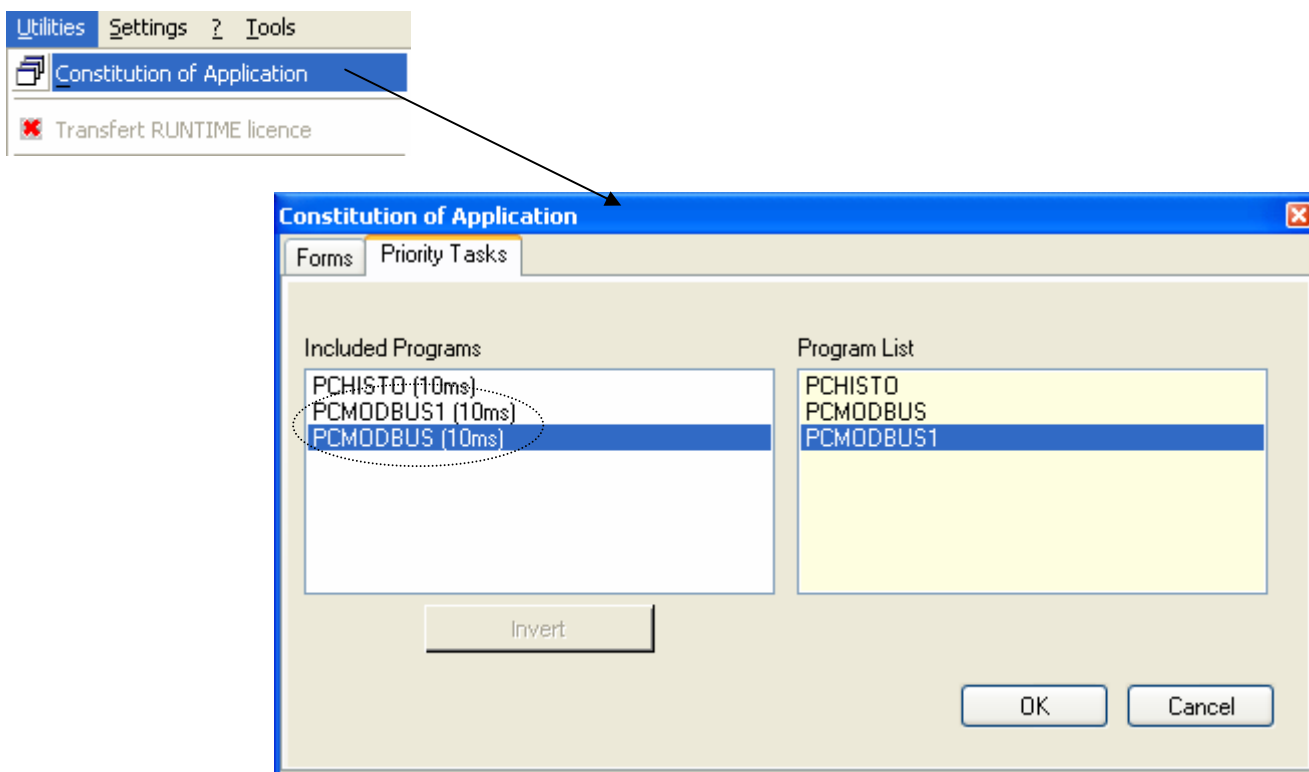
3/ Create a new configuration (on Com2)



You have now 2 Modbus drivers PCMODBUS1 and PCMODBUS running on 2 different comports.

Note if you need 3 Modbus drivers, rename the last PCmodbus in PCMODBUS2 and repeat the manipulation describe before.

Let's see the constitution of application to verify the presence of the 2 Modbus drivers in the current appli.



In the list of priority tasks there are the 2 drivers PCModbus and PCModbus1.