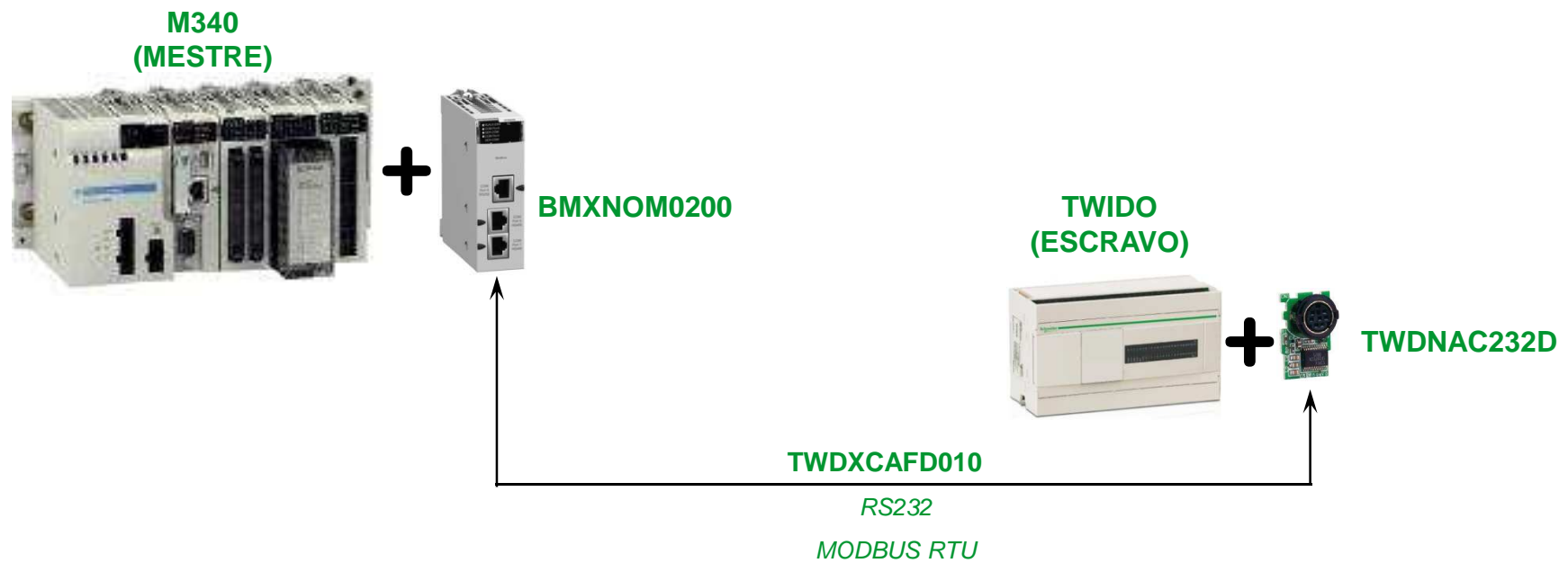


M340 + BMXNOM0200

Exemplo de comunicação *Modbus* RS232 entre M340 e TWIDO, utilizando BMXNOM0200.

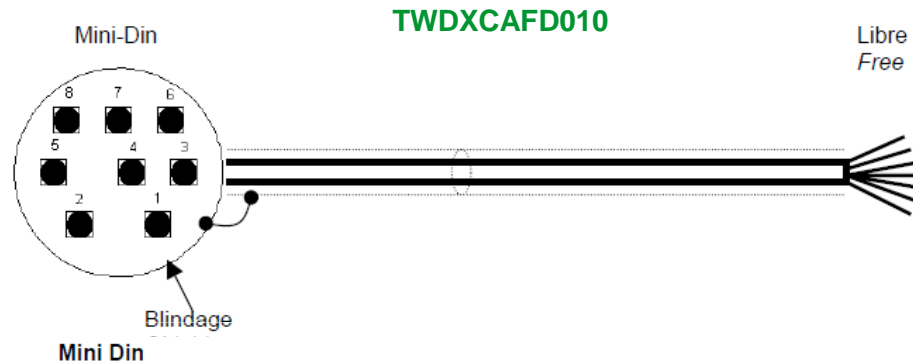
Comunicação

- Arquitetura da rede



Comunicação RS232

- Esquema de ligação



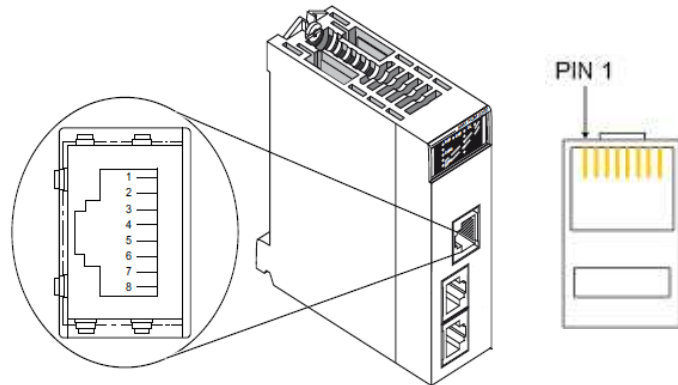
TWIDO RS232	TWIDO RS485	PIN	Couleur	Color
RTS	A+	1	Noir	Black
DTR	B-	2	Vert	Green
TXD	NC	3	Rouge	Red
RXD	/DE	4	Orange	Orange
DSR	/DPT	5	Jaune	Yellow
0V	NC	6	Blanc	White
0V	0V	7	Bleu	Blue
5V (*)	5V (*)	8	Marron	Brown
		Blindage Shield		

(*) : 180mA au total : 180mA max cumulative

Longueur : length : 1m

Comunicação RS232

- Esquema de ligação

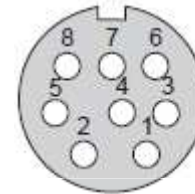


The table below shows the pin assignment for the serial port on the BMX NOM 0200:

Pin N°	RS232 channel 0 / port 0
1	RXD (Receive Data)
2	TXD (Transmit Data)
3	RTS (Request To Send)
4	DTR (Data Terminal Ready)
5	DSR (Data Set Ready)
6	CTS (Clear To Send)
7	DCD (Data Carrier Detect)
8	Potential serial link grounding (0 V)

TWDXCAFD010

Optional link
RS 232D
mini-DIN conn.



TWD NAC 232D
TWD NOZ 232D

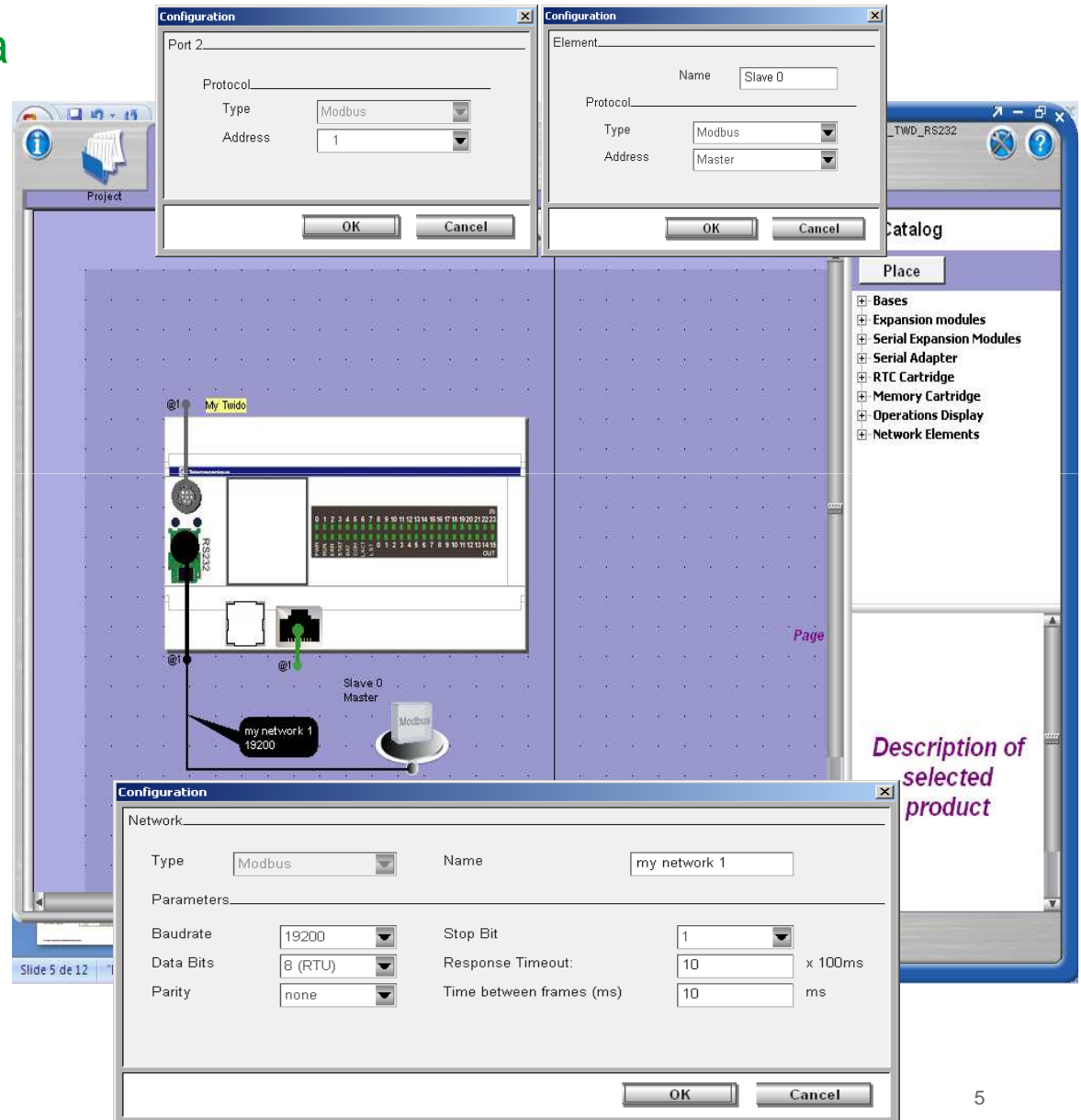
1	RTS
2	DTR
3	TXD
4	RXD
5	DSR
6	0 V
7	0 V
8	5 V (180 mA)



Twidosuite

- Configuração do programa

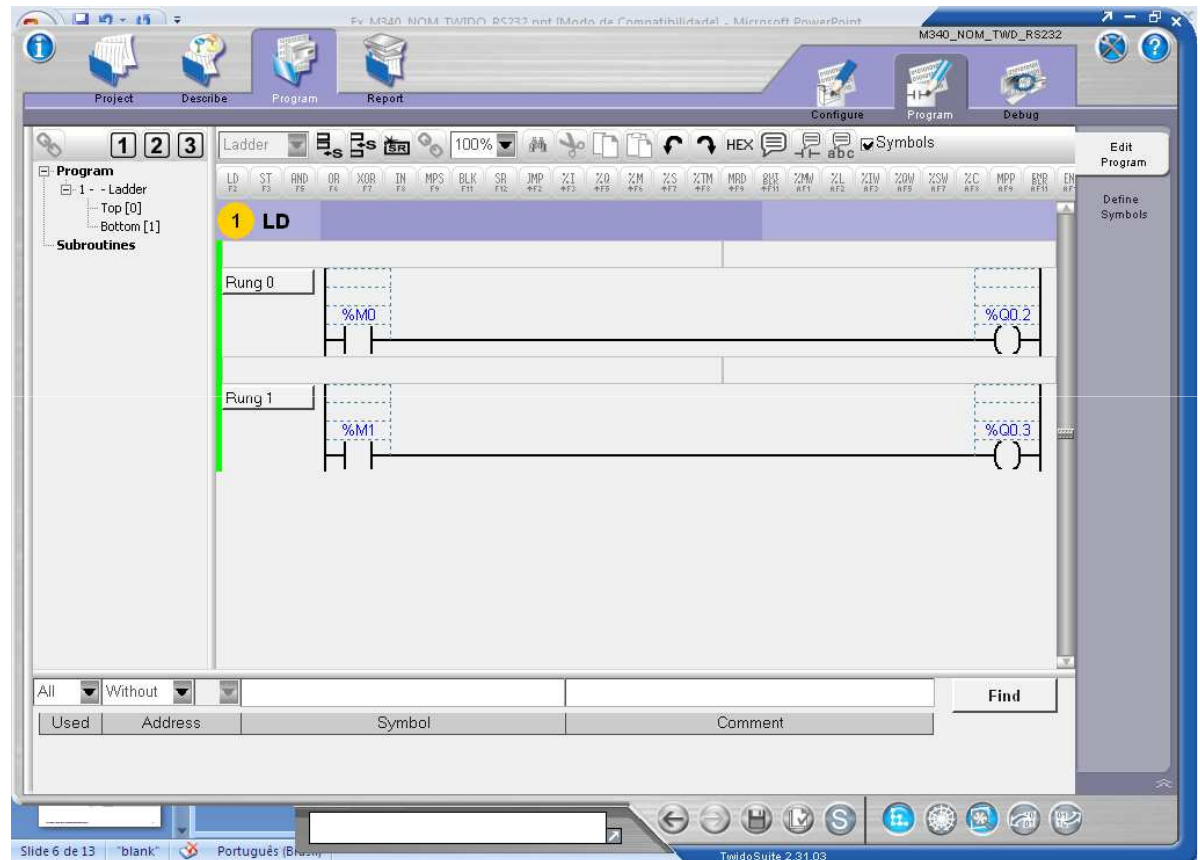
Para elaboração deste exemplo, foi utilizada uma CPU da Linha Twido, esta CPU foi declarada como escrava em uma rede *Modbus*, conforme ilustra figura ao lado.



Twidosuite

- Configuração do programa

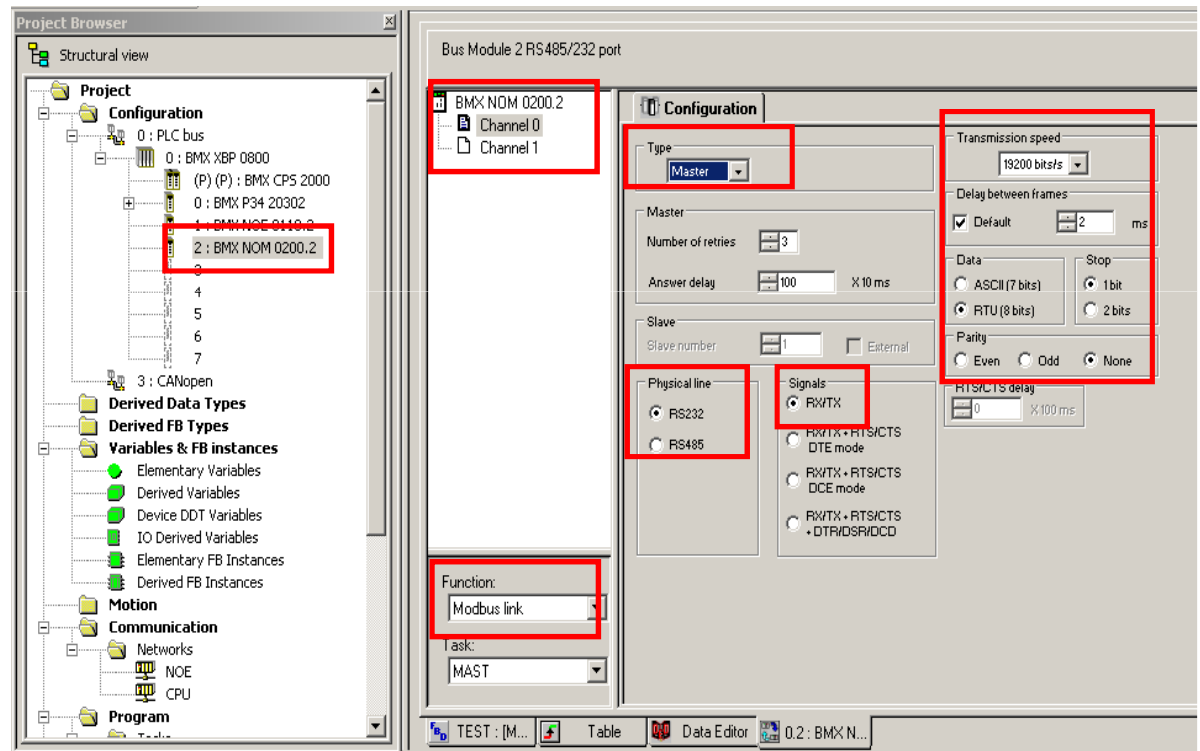
Neste exemplo, foram declaradas variáveis booleanas para serem lidas através da rede *Modbus*.



Unity Pro

- Configuração do programa

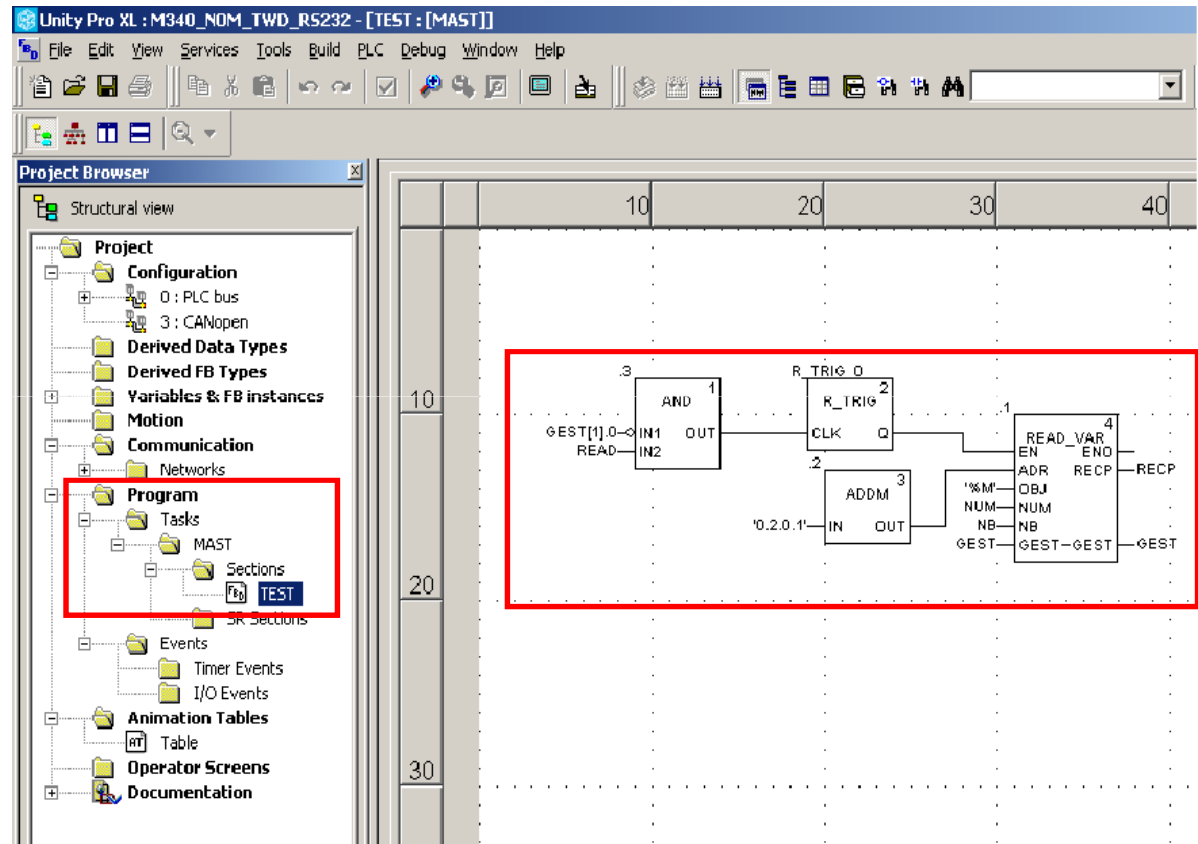
Na seqüência, configuramos o M340 onde, no Unity, deve-se realizar a declaração do *hardware* e efetuando duplo clique no módulo BMXNOM0200, teremos acesso as configurações do módulo. Neste exemplo ele foi configurado como mestre, protocolo *Modbus*, e utilizando somente os sinais TX/RX para o meio físico RS232



Unity Pro

- Configuração do programa

Após a configuração do cartão, foi criada uma seção de programa em FBD para elaboração da rotina de leitura do Twido. Para este exemplo, utilizou-se o bloco READ_VAR para leitura de memórias internas tipo %M do Twido, e o bloco ADDM, responsável pelo endereçamento para leitura do escravo. A seguir mais detalhes dos blocos ADDM e READ_VAR

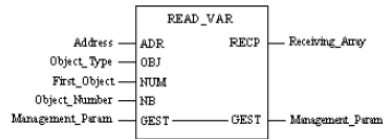


Unity Pro

• READ_VAR

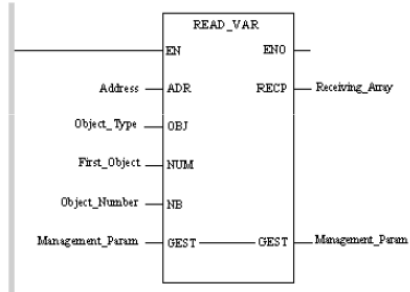
FBD representation

Representation:



.LD representation

Representation:



L representation

Representation:

LD Address
 READ_VAR Object_Type, First_Object, Object_Number, Management_Param, Receiving_Array

ST representation

Representation:

READ_VAR(Address, Object_Type, First_Object, Object_Number, Management_Param, Receiving_Array);

Function Description

The READ_VAR function is used to read the value of one or more language objects:

- internal bits,
- internal words.

The objects that are read must always be consecutive. They may be located in a remote CPU or in a device connected to a communication channel.

On Modicon M340 PLCs, the READ_VAR function can read up to 2,000 consecutive bits in a remote device.

Address	ARRAY [0.. 5] OF INT for Premium ARRAY [0.. 7] OF INT for Modicon M340	The following statements are right only for Premium PLCs: <ul style="list-style-type: none"> • Address of the message's receiving character mode channel is given by the ADDR function • Address of the destination entity of the exchange. The following addresses are prohibited: <ul style="list-style-type: none"> ◦ {Network.Station}APP, ◦ {Network.Station}APP.num. The following statements are right only for Modicon M340 PLCs: <ul style="list-style-type: none"> • Address of the message's receiving character mode channel is given by the ADDM function • The syntax of the address is of ADDM ('r.m.c.node')-type.
Object_Type	STRING	Type of objects to read for Premium PLCs: <ul style="list-style-type: none"> • '%M': internal bits, • '%MW': internal words, • '%S': system bits, • '%SW': system words, • '%I': input bits, • '%IW': input words. Type of objects to read for Modicon M340 PLCs: <ul style="list-style-type: none"> • '%M': internal bits, • '%MW': internal words,
First_Object	DINT	Index of the first object to read.
Object_Number	INT	Number of objects to read.

The following table describes the input/output parameters:

Parameter	Type	Comment
Management_Param	ARRAY [0.. 3] OF INT	Exchange management table On Modicon M340 PLCs, a new cancel bit is available in the rank 1 word of the exchange management table. This cancel bit is located at the rank 1 word which consists of 2 bytes: <ul style="list-style-type: none"> • Most significant byte: exchange number • Least significant byte: activity bit (rank 0) and cancel bit (rank 1). The READ_VAR EF can be cancelled by the CANCEL EF or by setting to 1 the cancel bit of the management table.

The following table describes the output parameters:

Parameter	Type	Comment
Receiving_Array	ARRAY [n... m] OF INT	Word table containing the value of the objects read.

Unity Pro

• READ_VAR - Observações

Leitura de bits devem seguir as considerações ao lado descritas:

At a Glance

On Modicon M340 PLCs, forced bits are not accessed by READ_VAR and WRITE_VAR functions as Modbus protocol does not support it. Therefore the following statements about forced bits cannot be applied to Modicon M340 PLCs.

The following examples illustrate the READ_VAR communication function for reading bits. In this case the receive table consecutively contains the value of the bits as well as the forcing indication.

Reading 32 internal bits

In ST, the syntax of the internal bit reading function is as follows:

```
READ_VAR (ADDR ('{20.1}0.5.1.3'), '%M', 0, 32, %MW100:4, %MW50:4);
```

The receive table must include 8 bytes (4 words), 4 bytes for the value and 4 bytes for the forcing indication.

	Word	Byte 3	Byte 2	Byte 1	Byte 0
Value	%MW100	0000	0000	1100	1100
	%MW101	1111	1111	0000	1111
Forcing	%MW102	0000	0000	0101	0101
	%MW103	0000	0000	0000	1111

Forcing can be performed for each %MW102 or %MW103 bit set to 1; the forcing value being that of the corresponding bit read.

Example:

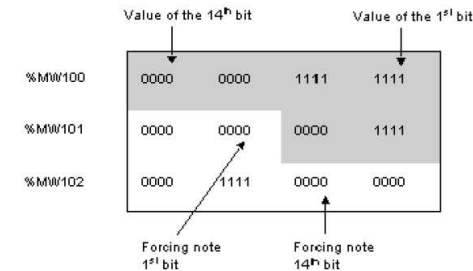
Value of the first 4 bits (byte 0 of word %MW100)	Forcing of the first 4 bits (byte 0 of word %MW102)	Description
0	1	The bit is forced to 0
0	0	The bit is not forced
1	1	The bit is forced to 1
1	0	The bit is not forced

Reading 18 internal bits

In ST, the syntax of the internal bit reading function is as follows:

```
READ_VAR (ADDR ('{20.1}0.5.1.3'), '%M', 0, 18, %MW100:3, %MW50:4);
```

The receive table must contain 3 words (or 6 bytes). In fact, to obtain the value of the 18 bits, 3 bytes plus a further 3 bytes are needed in order to contain the forcing value of the 18 bits.



Unity Pro

• ADDM

No programa exemplo, foi utilizada a sintaxe “r.m.c {host addr}” que indica o rack, a posição do módulo BMXNOM0200 no rack, canal do cartão que foi utilizado e o endereço do escravo. Estes detalhes podem ser visualizados na configuração do rack conforme ilustrado a seguir.

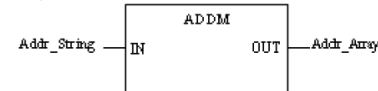
Function Description

The ADDM function is used to convert a character string into an address that can be used directly by the following communication functions: READ_VAR, WRITE_VAR, INPUT_CHAR, PRINT_CHAR, DATA_EXCH, SEND_EMAIL.

The additional parameters [EN](#) and [ENO](#) can be configured.

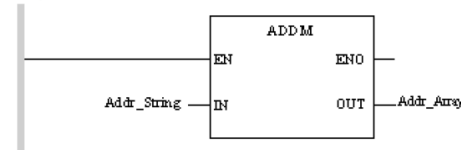
BD Representation

Representation:



LD Representation

Representation:



L Representation

Representation:

```
LD Addr_String  
ADDM  
ST Addr_Array
```

ST Representation

Representation:

```
Addr_Array := ADDM(Addr_String);
```

Parameter Description

The following table describes the input parameter:

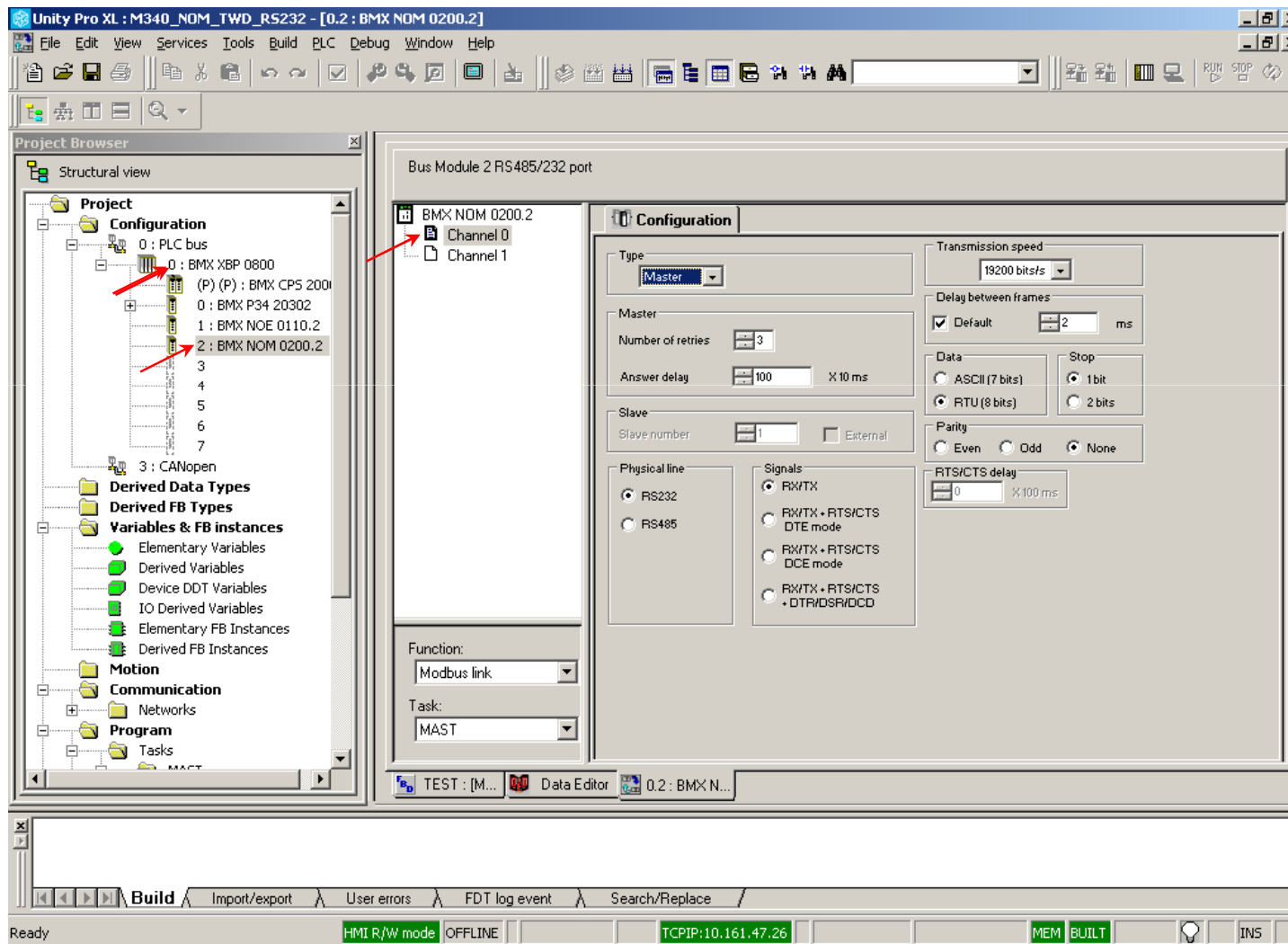
Parameter	Type	Comment
IN	Character string	Device address on bus or network.

For addressing a station on Ethernet, parameter IN takes the form:

- 'Netlink{hostAddr}'
- 'Netlink{hostAddr}TCP.MBS'
- 'Netlink{hostAddr}node'
- 'r.m.c{hostAddr}'
- 'r.m.c{hostAddr}TCP.MBS'
- 'r.m.c{hostAddr}node'
- '{hostAddr}'
- '{hostAddr}TCP.MBS'
- '{hostAddr}node'

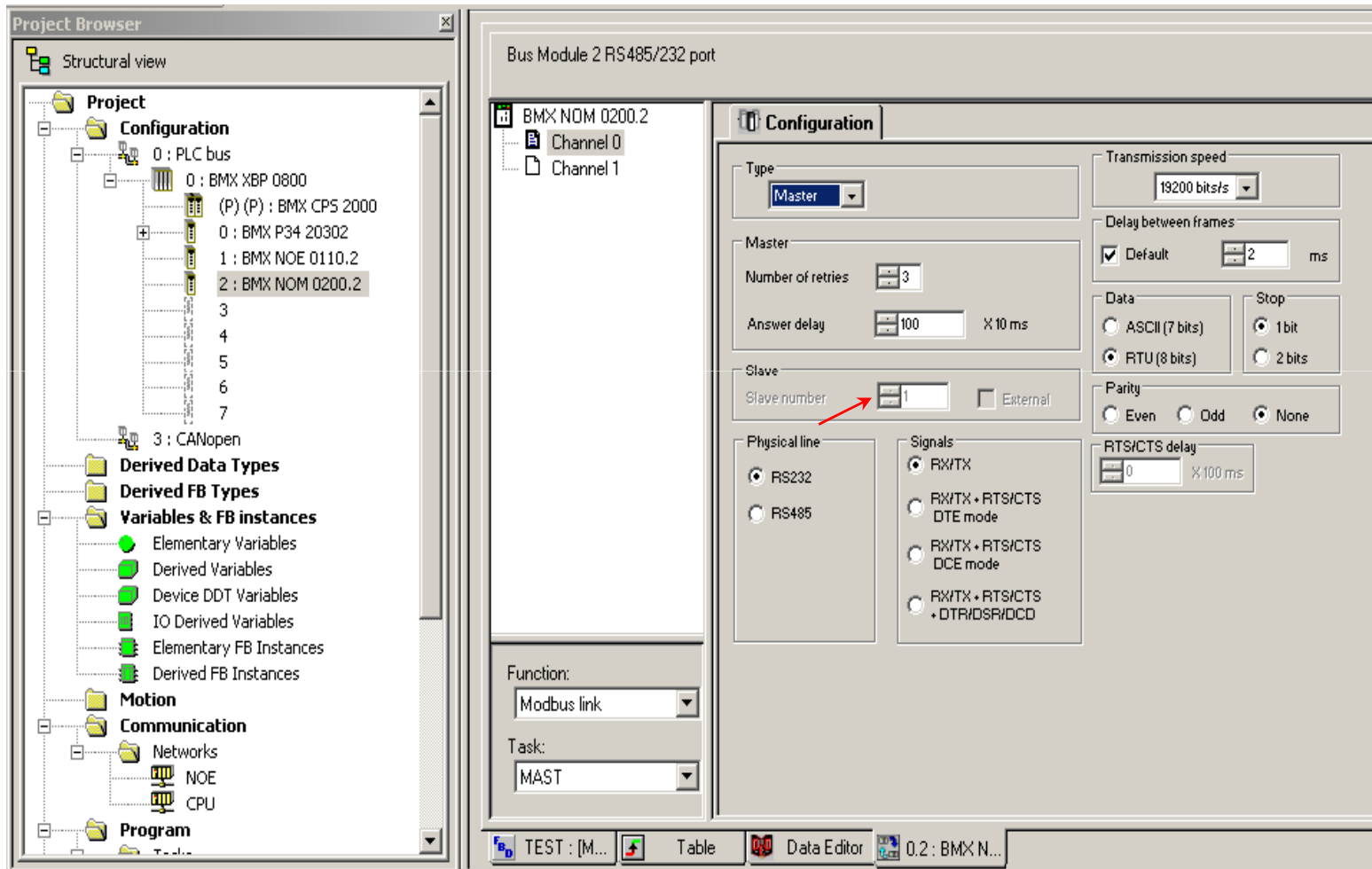
Unity Pro

- ADDM



Unity Pro

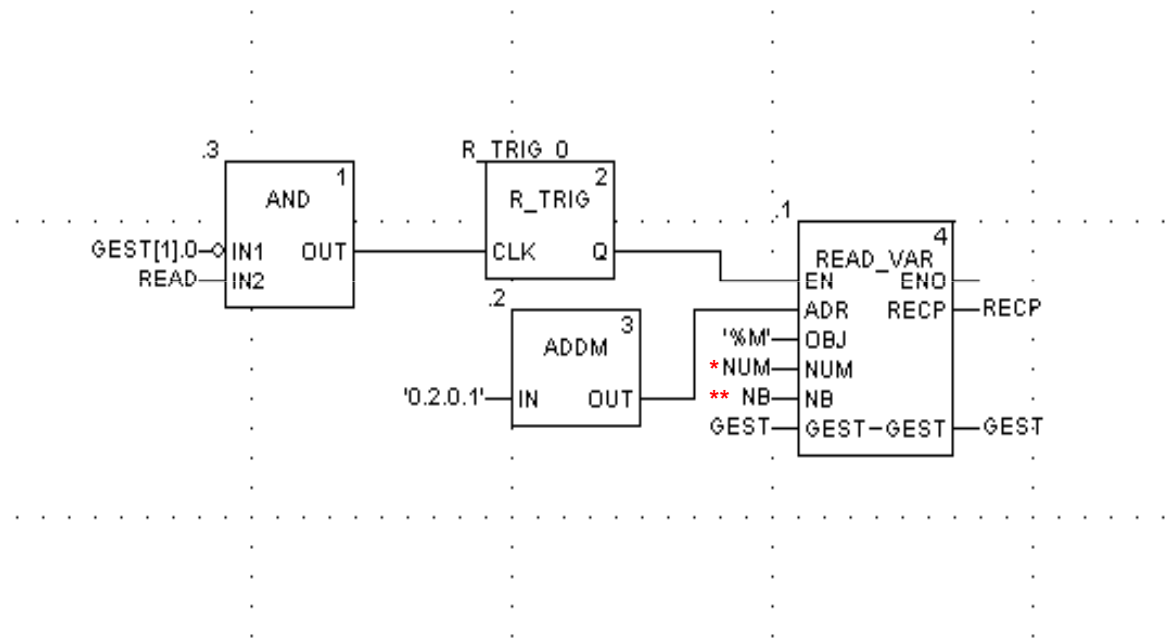
- ADDM



Unity Pro

- Configuração do programa

No programa exemplo, foi utilizado o Bit 0 do primeiro registro do Array da palavra de gestão “GEST”, ele tem por finalidade indicar a atividade da porta, habilitando o envio da mensagem somente quando a porta estiver disponível. Este é um modo de elaborar o gerenciamento de múltiplas mensagens (vários blocos de leitura/escrita) que deve ser desenvolvido pelo usuário.



* Endereço do primeiro bit a ser lido
** Quantidade de bits a serem lidos

Informações adicionais

Este documento é um complemento, que deve ser usado como apoio e em nenhum momento substitui as informações contidas no manuais de cada produto. Para maiores informações ou esclarecimentos:

www.schneider-electric.com.br

- Call Center

- Tel: 0800 7289 110 ou (11) 3468-5791
- E-mail: call.center.br@br.schneider-electric.com

- Centro de Treinamento

- Consulte o calendário de treinamentos pelo site ou para maiores informações:
- Tel: (11) 2165-5350
- E-mail: treinamento.br@br.schneider-electric.com