

Technical Guide

PLCnext Engineer

Como transferir uma variável do tipo **REAL** através de uma comunicação Modbus ou Profinet.



PLCnext Technology 
Designed by PHOENIX CONTACT



PLCnext Control



PLCnext Engineer



PLCnext Store



PLCnext Community

INTRODUÇÃO

A comunicação MODBUS utiliza como padrão a transferência de variáveis do tipo WORD. Já a comunicação PROFINET a troca de dados é feita através de variáveis do tipo BYTE.

Portanto a transferência de variáveis do tipo REAL, requer uma preparação, quebrando esta variável em WORD ou BYTES, transmitindo e posteriormente é necessário reconstruí-la. Alguns equipamentos fazem isso de forma automática, porém quando realizamos a comunicação entre dois controladores esse preparação é necessária.

INTRODUÇÃO

- Variável do tipo REAL = 32 Bits
- Variável do tipo DWORD = 32 Bits
- Variável do tipo WORD = 16 Bits
- Variável do tipo BYTE = 8 Bits

DWORD																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WORD 1																WORD 0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BYTE 3								BYTE 2								BYTE 1								BYTE 0							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Criar um novo projeto no PLCnext Engineer



PLCnext Engineer Engineering Software

Version 2021.0 (Build 4.7.1137.0)

Copyright 2020
PHOENIX CONTACT GmbH & Co. KG
32823 Blomberg

For up-to-date software versions
and error removals, visit
www.phoenixcontact.com



Selecione o modelo do controlador e sua versão de firmware

Start Page





PHOENIX CONTACT
INSPIRING INNOVATIONS

PLCnext Engineer 2021.0

(Build 4.7.1137.0)

Recent projects
C:\Users\Public\...\Projects\PROJETO_0_2021.pcwex

Project templates

-  AXC F 1152 v00 / 2020.6.0
-  AXC F 1152 v00 / 2021.0.0
-  AXC F 2152 v00 / 2020.6.0
-  AXC F 2152 v00 / 2021.0.0

Need help?

Welcome to PLCnext Engineer
The Start Page welcomes you introducing PLCnext Engineer. The Start Page is shown every time you start PLCnext Engineer and is automatically closed when opening or creating a project.

Recent projects
The 'Recent projects' list shows the projects that you have recently opened in PLCnext Engineer. Click the project name to open the desired project.

Project templates
The 'Project templates' list provides a list of predefined project templates to help you get started. Selecting a project template creates a new PLCnext Engineer project with the desired controller type already added.

Further help
If you do not know what to do after a project is opened, you can find further help [here](#).

Criar o FB(Function Block) que irá quebrar a variável REAL em 4 BYTES

The screenshot displays the Siemens SIMATIC Manager interface. On the left, the 'PLANT' tree shows a project named 'axc-f-2152-1 : AXC F 2152' containing components like PLC, HMI Webserver, OPC UA, Profinet, and Axioline F. On the right, the 'COMPONENTS' tree shows a 'Programming (289)' folder with a 'Local (1)' sub-folder containing 'Data Types (0)', 'Functions & Function Blocks (0)', 'Programs (1)', 'Extended (72)', 'IEC 61131-3 (125)', 'PLCnext Controller (35)', and 'Safety IEC 61131-3 (56)'. A context menu is open in the center, with 'Add Function Block' highlighted in red. Other menu items include 'Add Function', 'Add SFC Function Block', 'Add Compact SFC Function Block', 'Add Safe FunctionBlock', 'Add Safe C FunctionBlock', 'Add Folder', and 'Paste (Ctrl+V)'. The bottom toolbar shows various icons for navigation and editing.

Criar o FB(Function Block) que irá quebrar a variável REAL em 4 BYTES

FB_REAL_TO_4_BYTES x

Select the programming language of your first worksheet below

```
1 IF condition = TRUE THEN
2   opC := opA AND opC;
3 END_IF
```

Add ST Code Worksheet

Add LD Code Worksheet

Network (1) Network One

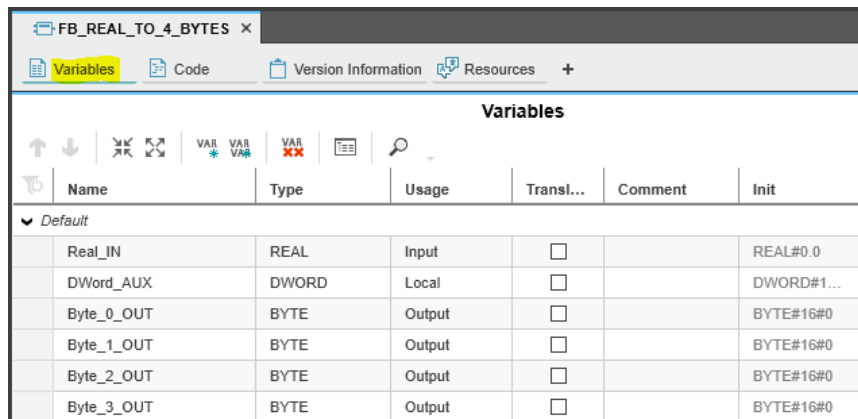
Add NOLD Code Worksheet

COMPONENTS

Search

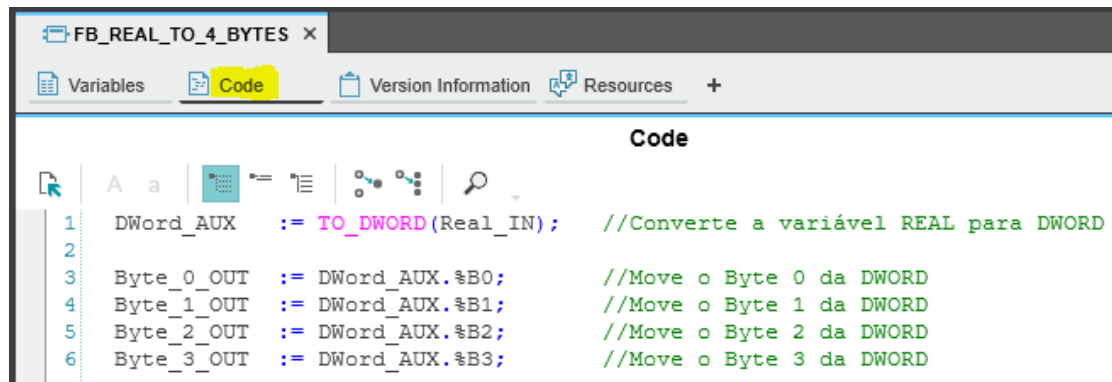
- Programming (290)
 - Local (2)
 - Data Types (0)
 - Functions & Function Blocks (1)
 - FB_REAL_TO_4_BYTES
 - Programs (1)
 - Extended (72)
 - IEC 61131-3 (125)
 - PLCnext Controller (35)
 - Safety IEC 61131-3 (56)

Criar o FB(Function Block) que irá quebrar a variável REAL em 4 BYTES



The screenshot shows the 'Variables' tab of a function block editor. The interface includes a toolbar with icons for variable declaration and search, and a table listing the variables. The table has columns for Name, Type, Usage, Transl..., Comment, and Init.

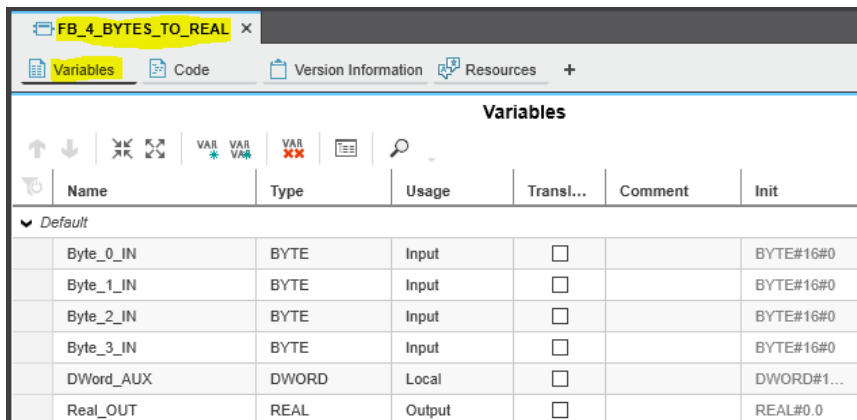
Name	Type	Usage	Transl...	Comment	Init
Real_IN	REAL	Input	<input type="checkbox"/>		REAL#0.0
DWord_AUX	DWORD	Local	<input type="checkbox"/>		DWORD#1...
Byte_0_OUT	BYTE	Output	<input type="checkbox"/>		BYTE#16#0
Byte_1_OUT	BYTE	Output	<input type="checkbox"/>		BYTE#16#0
Byte_2_OUT	BYTE	Output	<input type="checkbox"/>		BYTE#16#0
Byte_3_OUT	BYTE	Output	<input type="checkbox"/>		BYTE#16#0



The screenshot shows the 'Code' tab of the function block editor. The code is written in a ladder logic style with line numbers 1 through 6. The code implements the conversion of a REAL variable to a DWORD and then extracts its four bytes into separate output variables.

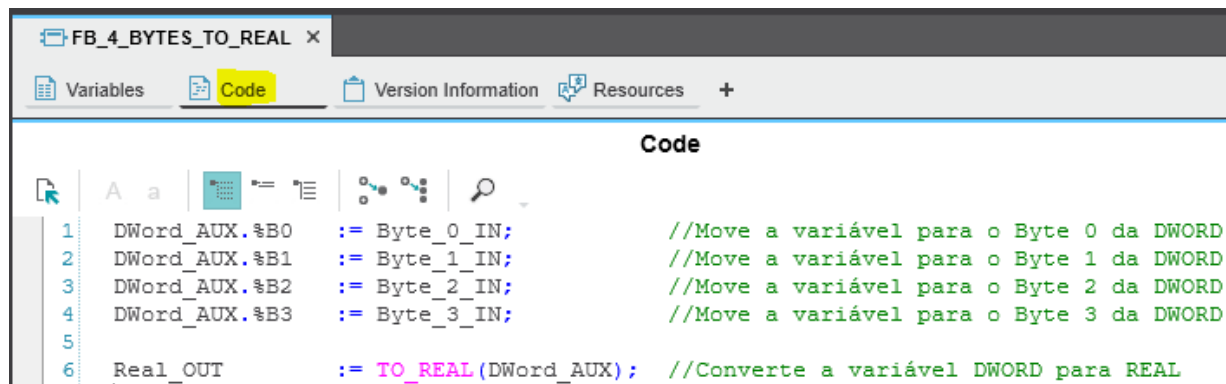
```
1 DWord_AUX := TO_DWORD(Real_IN); //Converte a variável REAL para DWORD
2
3 Byte_0_OUT := DWord_AUX.B0; //Move o Byte 0 da DWORD
4 Byte_1_OUT := DWord_AUX.B1; //Move o Byte 1 da DWORD
5 Byte_2_OUT := DWord_AUX.B2; //Move o Byte 2 da DWORD
6 Byte_3_OUT := DWord_AUX.B3; //Move o Byte 3 da DWORD
```


Criar o FB(Function Block) que irá reconstruir 4 BYTES em uma variável REAL



The screenshot shows the 'Variables' tab for the function block 'FB_4_BYTES_TO_REAL'. It displays a table of variables with columns for Name, Type, Usage, Transl..., Comment, and Init.

Name	Type	Usage	Transl...	Comment	Init
▼ Default					
Byte_0_IN	BYTE	Input	<input type="checkbox"/>		BYTE#16#0
Byte_1_IN	BYTE	Input	<input type="checkbox"/>		BYTE#16#0
Byte_2_IN	BYTE	Input	<input type="checkbox"/>		BYTE#16#0
Byte_3_IN	BYTE	Input	<input type="checkbox"/>		BYTE#16#0
DWord_AUX	DWORD	Local	<input type="checkbox"/>		DWORD#1...
Real_OUT	REAL	Output	<input type="checkbox"/>		REAL#0.0



The screenshot shows the 'Code' tab for the function block 'FB_4_BYTES_TO_REAL'. The code is as follows:

```
1 DWord_AUX.B0 := Byte_0_IN; //Move a variável para o Byte 0 da DWORD
2 DWord_AUX.B1 := Byte_1_IN; //Move a variável para o Byte 1 da DWORD
3 DWord_AUX.B2 := Byte_2_IN; //Move a variável para o Byte 2 da DWORD
4 DWord_AUX.B3 := Byte_3_IN; //Move a variável para o Byte 3 da DWORD
5
6 Real_OUT := TO_REAL(DWord_AUX); //Converte a variável DWORD para REAL
```

Instanciar os FBs criados no programa “Main”

Select the programming language of your first worksheet below

```
1 IF condition = TRUE THEN
2   opC := opA AND opC;
3 END_IF
```

Add ST Code Worksheet

Add LD Code Worksheet

Network (1) Network One

Add NOLD Code Worksheet

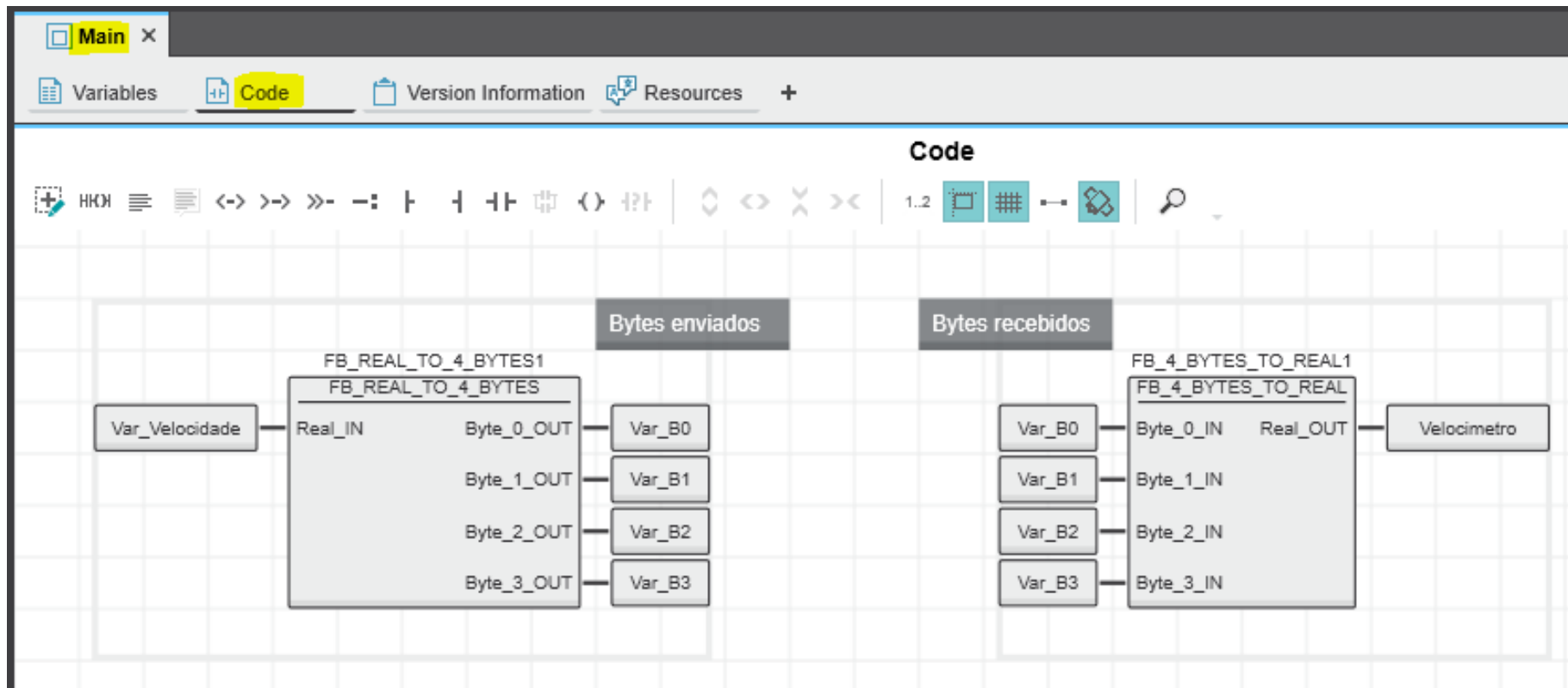
Close

COMPONENTS

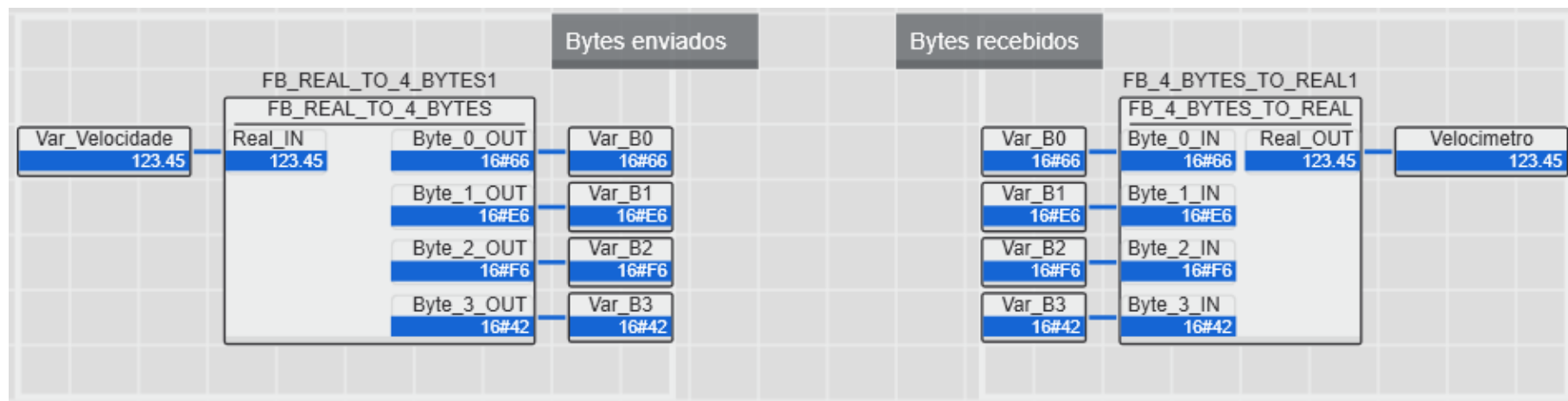
Search

- Programming (291)
 - Local (3)
 - Data Types (0)
 - Functions & Function Blocks (2)
 - FB_4_BYTES_TO_REAL
 - FB_REAL_TO_4_BYTES
 - Programs (1)
 - Main
 - Extended (72)
 - IEC 61131-3 (125)
 - PLCnext Controller (35)
 - Safety IEC 61131-3 (56)

Instanciar os FBs criados no programa “Main”



Em modo Degub “Online”



Pronto!!!

Os próximos passos referentes a configurar da comunicação MODBUS ou PROFINET, estão detalhados em outros guias, que tratando especificamente destas comunicações.

Apresentação



Rodrigo Richter

Eng. de aplicação e desenvolvimento de negócios.

rrichter@phoenixcontact.com.br

Obrigado...



PLCnext Technology[®]
Designed by PHOENIX CONTACT